AFRL-VA-WP-TR-2000-3009

DEVELOPMENT AND VALIDATION OF PARALLEL DISTRIBUTED COMPUTING ENVIRONMENT FOR AEROSTRUCTURAL CFD ANALYSIS

VINCENT J. HARRAND
VIJI PARTHASARATHY
ESSAM F. SHETA
CHARLES W. WARREN
MARK L. UNDERWOOD

CFD RESEARCH CORPORATION
215 WYNN DRIVE
HUNTSVILLE, AL 35805

FEBRUARY 2000

FINAL REPORT FOR MAY 1996 – OCT 1999

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

AIR VEHICLES DIRECTORATE
AIR FORCE RESEARCH LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE OH 45433-7542

NOTICE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

THIS TECHNICAL REPORT HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION.

Don W. Kinsey
DON W. KINSEY
Chief
Computational Sciences Branch

Douglas L. Bowers
DOUGLAS L. BOWERS
Chief
Aeronautical Sciences Division

Do not return copies of this report unless contractual obligations or notice on a specific document requires its return.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	FEBRUARY 2000	FINAL REPORT FOR MAY 1996 - OCT 1999

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
DEVELOPMENT AND VALIDATION OF PARALLEL DISTRIBUTED COMPUTING ENVIRONMENT FOR AEROSTRUCTURAL CFD ANALYSIS	C F33615-96-C-3002 PE 62201 PR 2404 TA 10 WU TN
6. AUTHOR(S) VINCENT J. HARRAND, CHARLES W. WARREN VIJI PARTHASARATHY, MARK L. UNDERWOOD ESSAM F. SHETA	

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
CFD RESEARCH CORPORATION 215 WYNN DRIVE HUNTSVILLE, AL 35805	A-9812

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AIR VEHICLES DIRECTORATE AIR FORCE RESEARCH LABORATORY AIR FORCE MATERIEL COMMAND WRIGHT-PATTERSON AFB, OH 45433-7542 POC: DON W. KINSEY, AFRL/VAAC, 937-255-4305	AFRL-VA-WP-TR-2000-3009

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE
APPROVED FOR PUBLIC RELEASE, DISTRIBUTION UNLIMITED.	

13. ABSTRACT *(Maximum 200 words)*

Prediction and control of aeroelastic phenomena is a complex multi-disciplinary problem due to the interaction of aerodynamic, elastic and inertia forces on the aircraft structure. Lack of integration among computer programs that serve different disciplines, such as aerodynamics and structures,has posed a major obstacle to accurate aeroelastic analysis. The Multi-Disciplinary Computing Environment (MDICE) has been developed to help solve this problem. MDICE enables engineering analysis codes to perform coupled multi-diciplinary analysis in a distributed computing environment. A unique feature is that existing engineering analysis codes are being used with a high level of interoperability and interchangeability.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Computational Fluid Dynamics, Computational Aeroelasticity, Computing Environments	65
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

Standard Form 298 (Rev. 2-89) (EG)
Prescribed by ANSI Std. 239.18
Designed using Perform Pro, WHS/DIOR, Oct 94

TABLE OF CONTENTS

		Page
	FOREWARD	
1.	INTRODUCTION	1
2.	PROJECT OBJECTIVES	3
3.	MDICE SYSTEM OVERVIEW	6
4.	PROJECT OVERVIEW	15
	4.1 Development of MDICE Computational Environment	15
	4.2 Integration of Engineering Analysis Codes	19
	4.3 Development of MDICE Modules	22
	4.4 Demonstration and Validation Cases	24
5.	COLLABORATION WITH AEROSPACE COMPANIES AND UNIVERSITIES	55
	5.1 Northrop Grumman	55
	5.2 Lockheed Martin	57
6.	RECOMMENDATIONS FOR FUTURE WORK	59
7.	REFERENCES	62
8.	MDICE BIBLIOGRAPHY	63

LIST OF FIGURES

Page

Figure 1. Conceptual Overview of the Multi-Disciplinary Computing Environment MDICE 9
Figure 2. GUI Options for MDICE 11
Figure 3. Conceptual Overview of MDICE-AE 12
Figure 4. Two Examples of the MDICE Structural Solver GUI 24
Figure 5. Computational CFD Mesh for AGARD Wing 27
Figure 6. Steady State Solution with Leading and Trailing Edge Displacement . . . 28
Figure 7. Original vs. Interpolated Mode Shapes 29
Figure 8. First Two Mode Shapes (Flutter) 30
Figure 9. First Two Mode Shapes (Original vs. Interpolated Mode Shapes) 31
Figure 10. Overview of Original vs. Interpolated Mode Shapes for Flutter Calculations 32
Figure 11. First Two Mode Shapes of Viscous Flutter Calculation 32
Figure 12. Original vs. Interpolated Mode Shapes for Viscous Flutter Simulation . . 33
Figure 13. Roe vs. Van Leer Sensitivity on Flutter Simulation 34
Figure 14. Time Step Sensitivity on Flutter Simulation 35
Figure 15. Computational CFD Grid 37
Figure 16. Surface Pressure Contours on Deflected Geometry 38
Figure 17. Wing Tip and Midspan Displacement as a Function of the Number of Fluid-Structure Exchanges 38
Figure 18. Computational CFD Grid 40
Figure 19. Iso Total Pressure Surfaces for Rigid Model 41
Figure 20. Iso Total Pressure Surfaces for Flexible Model 41
Figure 21. Root Bending Moment and Tail-Tip Bending and Torsion Deflections . 42
Figure 22. Twin Tail Configuration 44
Figure 23. Iso-Valve Surfaces of Total Pressure 46
Figure 24. Total Pressure Iso-Surfaces as a Function of Angle of Attack 47
Figure 25. Tail-tip Bending and Twisting Deflection for Left and Right Tail 48
Figure 26. Root Bending Moment for Various Angles of Attack 49
Figure 27. Mean and RMS Valves of the Right Tail Root Bending Coefficients as a Function of the Angle of Attack 50
Figure 28. Lift Coefficient as a Function of Angle of Attack 51
Figure 29. RMS Buffet and Surface Pressures at Five Transducer Locations on the Right Tail 52
Figure 30. Buffet Excitation Spectra 53
Figure 31. First Two Dominant Non-Dimensional Frequencies as a Function of Angle of Attack 54

FOREWORD

This report summarizes all project related issues for the "Development and Validation of Parallel Distributed Computing Environment for Aerostructural CFD Analysis" – project. One of the main results of this project is the software code MDICE (Multi-Disciplinary Computing Environment). All information directly related to the software has been documented in three accompanying software manuals, in particular:

1. MDICE Multi-Disciplinary Computing Environment, User's Guide, Version 6.0, October 1999.
2. MDICE-AE, User's Guide, October 1999.
3. MDICE Programmer's Guide, October 1999.

Besides reading this report, the reader is encouraged to read the software manuals if a greater amount of technical detail is required. Furthermore, an MDICE bibliography has been included at the end of this report.

This program has been funded by the AFRL/VA Directorate under contract number F33615-96-C-3002. Technical Monitors from AFRL have been Capt. Joel Luker, Dr. Don Kinsey (VAAC), and Mr. Larry Huttsell (VASD). During the course of the project we have been collaborating with many people at AFRL, Universities, and all major aerospace companies. We would like to thank all people involved for their contributions, in particular John Volk, Steve Brown, Charley Peavey, Ed Blosch (Northrop Grumman), Mike Love, Tony Delagarza, Eric Charlton (LMTAS), Rudy Yurkovich, Rob Rattcliff

(Boeing), Ray Kolonay (GE), Erwin Johnson (MacNeal Schwendler), Marilyn Smith (Georgia Tech), Tom Strganac (Texas A&M), Rich Snyder, and Reid Melville (AFRL).

There have been very many people at CFD Research Corporation who have significantly contributed to this project. The authors would like to thank Ashok Singhal, Curtis Mitchell, Paul Dionne, Gerry Kingsley, John Siegel, Freddy Golos, Bill Coirier, David Fricker, John Whitmire, Vadim Uchitel, Stacy Rock, Sami Habchi, Denise Rynders, and Jennifer Swann for all their contributions.

It needs to be stressed that although this report mentions "Final Report", the MDICE related work is by no means final. The current contract will remain active and several other application oriented contracts are in place. CFD Research Corporation is very pleased with the significant interest in MDICE and related technologies from other USAF branches, NASA, and the aerospace companies.

Vincent Harrand

Program Manager

1. INTRODUCTION

One of the critical ongoing research programs in the military aerospace community is the fighter aircraft sustainment program. Under this program the DoD and other Governmental research centers are trying to significantly reduce aircraft and mission costs, decrease maintenance cost, and increase mission capabilities. The program has placed particular emphasis on improving the super-maneuverability of aircraft during combat. A critical limitation is the existence of several instability problems which limit the fighter's designed operating envelope and decreases its flight missions.

Prediction and control of aeroelastic phenomena are a very complex multi-disciplinary problem due to the interaction of aerodynamic, elastic, and inertia forces on the aircraft structure. Those forces produce the oscillation that often results in premature structure failure. However, many current problems, e.g. wing flutter, tail buffeting, and store induced limit cycle oscillations, involve several disciplines and require a truly multi-disciplinary simulation capability. Lack of integration among computer programs that serve different disciplines has posed a major obstacle to such analysis.

For this purpose, the Multi-Disciplinary Computing Environment (MDICE) has been developed as part of this project. MDICE enables engineering analysis codes to perform coupled multi-disciplinary analysis in a distributed computing environment. A unique feature is that existing engineering analysis codes are being used with a high level of interoperability and interchangeability. MDICE constitutes a new approach for multi-

disciplinary analysis and has enabled a significant step forward on solving an important class of multi-disciplinary problems.

2. PROJECT OBJECTIVES

The objective of this program was to conceptualize, develop, test, and validate a multi-disciplinary computing environment for Aerostructural CFD analysis. In spite of the commendable progress in CFD, structural analysis, 3D design optimization, geometry modeling, grid generation, scientific data visualization, and computer systems, the process of multi-disciplinary analysis is still very inefficient and inconvenient. The main deficiencies are rooted in the current practices of: data exchange (via files), data management (most manual), and software development (with redundancy and duplication of very many basic functions). Real world applications, such as aeroelastic analysis of air vehicles, often require the coupling of the aforementioned analysis techniques into one multi-disciplinary computing environment. Being able to perform such an analysis in a timely and efficient manner is crucial for impacting the design and analysis process for air vehicles. The efficiency goal puts a high emphasis on the architecture of the computational environment and the interfacing methods used for data exchange among the various codes.

The environment will provide:

- Open architecture (and hence flexibility) for using existing and new analysis codes;
- Full associativity (and hence precision) between geometry, grids (CFD and CSD) and data;
- Accurate and efficient interpolation methods;

- Enable a generic capability for volumetric grid movement/deformation.
- Efficient utilization of computer resources, particularly cluster of heterogeneous work stations; and
- Full user control of the simulation process via a graphical user interface.

The reliability and effectiveness of the system will be tested by using aeroelastic wing problems, and then further demonstrated by simulating two validation (body/wing and full body) problems.

The computational environment was to be tested with CFDRC codes, and later with AFRL and third-party engineering analysis codes. In order to ensure the quality of the software itself (i.e. usability, generality, ease of integration, etc.), a strong collaboration with AFRL and at least one airframe company was envisioned. At the start of the contract, Northrop Grumman (Pico Rivera, CA) was selected to collaborate on this contract. During the third year, Lockheed Martin Tactical Aircraft Systems (Forth Worth, TX) started to use and validate the software on their own behalf and provided valuable feedback to the project.

In the remainder of this report an overview of the entire project is given. The following topics are addressed:

- MDICE system overview,
- Overview of all project work
  - MDICE software development activities,

  - code integration overview and status,
  - new module development, and
  - testing/validation of environment,
- Collaboration with aerospace companies, and
- Recommendations for future work.

3. MDICE SYSTEM OVERVIEW

Throughout the years there have been basically two approaches to fluid-structure interaction, i.e. (1) a loosely coupled approach, and (2) a tightly coupled approach. In the loosely coupled approach the fluids and structural solutions are computed separately, and manually the data is converted from the CFD to the CSD program. In the tightly coupled approach the various modules, i.e. CFD, CSM, and interfacing, are in one executable and communicate directly. Technically, there is one more approach which is tightly coupled both from a physics and a program point of view (the aforementioned methods are loosely coupled from a physics point of view). CFDRC's CFD-ACE+ solver is an example of the latter category.

In the loosely coupled approach, the end-user process was fairly tedious and the data exchange was usually performed only once. The entire process may have taken weeks to accomplish. The FASIT Fluid-Structure Interface Program [1] has contributed significantly to this process by providing several fluid-structure interfacing methods with an easy to use graphical user interface. A steady state analysis process may now take 3 days with many data exchanges between the two codes. The advantage of the loosely coupled approach is that existing codes can be coupled and that the end-user can make a choice in selecting a code based on technical, familiarity and/or validation reasons.

In the tightly coupled approach, the end-user process is highly automated and no manual intervention is required. The disadvantage is that the code is large, and therefore difficult

to maintain. The various modules are fixed and can not be exchanged for other modules. For the various aerospace companies and their project offices, it may be difficult to accept new aeroelastic solvers because they have not been validated on their aircraft. In addition, different technologies, which may not be available in the integrated aeroelastic solver, may be needed to solve a particular problem.

The MDICE Multi-Disciplinary Computing Environment combines the strong points of both approaches into one environment. The fluid-structure interaction process is fully automated and there is a choice of solvers.

MDICE enables steady-state simulations with only a few interactions and/or transient simulations with thousands of interactions. This creates a highly <u>efficient</u> end-user process. Combined with the distributed nature of the environment, in which all codes could run on a different computer platform, a steady state simulation may now take a few hours to compute. In addition, multi-level parallelism is supported allowing parallelized flow solvers to run in parallel with a structural analysis program. By means of a graphical user interface, the user labels the CFD patch needed to communicate with the structural patch. This is all the end user has to do in addition to setting up the input decks to the structural and flow solvers (which usually pre-exist).

The MDICE end-user can be very <u>effective</u>. The best modules can be selected for the particular case at hand, based on familiarity or technical reasons. Different flow solvers can be selected which support structured grids, unstructured grids, or polyhedral grids.

Different structural solvers for influence coefficients, modal analysis, specialized beam models, linear finite elements, and non-linear finite elements are available. Moreover, there are several fluid-structure interfacing methods available within MDICE.

The MDICE environment offers a great deal of flexibility to the end user. One module can be transparently replaced with another module by the end user. In addition, MDICE can be operated in a heterogeneous computing environment, from a PC (NT, 98, Linux), to a UNIX workstation, to a multi-processor supercomputer under a batch queuing system (or a combination thereof).

The MDICE environment has a very general architecture and all interactions between modules are modeled by means of objects. This ensures that a high degree of extensibility is obtained. New modules can be added by direct integration, i.e. invoke the MDICE API (set of functions) from the module source code. Alternatively, a generic wrapper exists for file-based integration of modules of which only executable code is available. New disciplines are easily added to the system by using existing or adding new objects. For example, the MDICE software is currently being used in areas such diverse as external aerodynamics, propulsion, biomedical and electronics manufacturing applications.

Figure 1 shows a conceptual overview of the Multi-Disciplinary Computing Environment MDICE. Several engineering disciplines are supported by MDICE and have been demonstrated on a variety of applications. Other disciplines, including thermal, electromagnetics, controls, trajectory, and optimization can be added in the future. For

each discipline, one or more engineering analysis codes have been integrated. Besides analysis oriented codes, typical design-oriented codes can be integrated as well. MDICE has several 'zooming' methods available for coupling low-fidelity/dimensionality applications with higher fidelity applications. Furthermore, spreadsheet oriented cost or performance based models and data have been interfaced and demonstrated with MDICE. In particular, interfacing MDICE with Product Data Management (PDM) databases would be the next logical step for large corporations. Several example MDICE-compliant engineering analysis codes are shown in Figure 1. A detailed overview of all MDICE compliant applications is given in section 4.2.

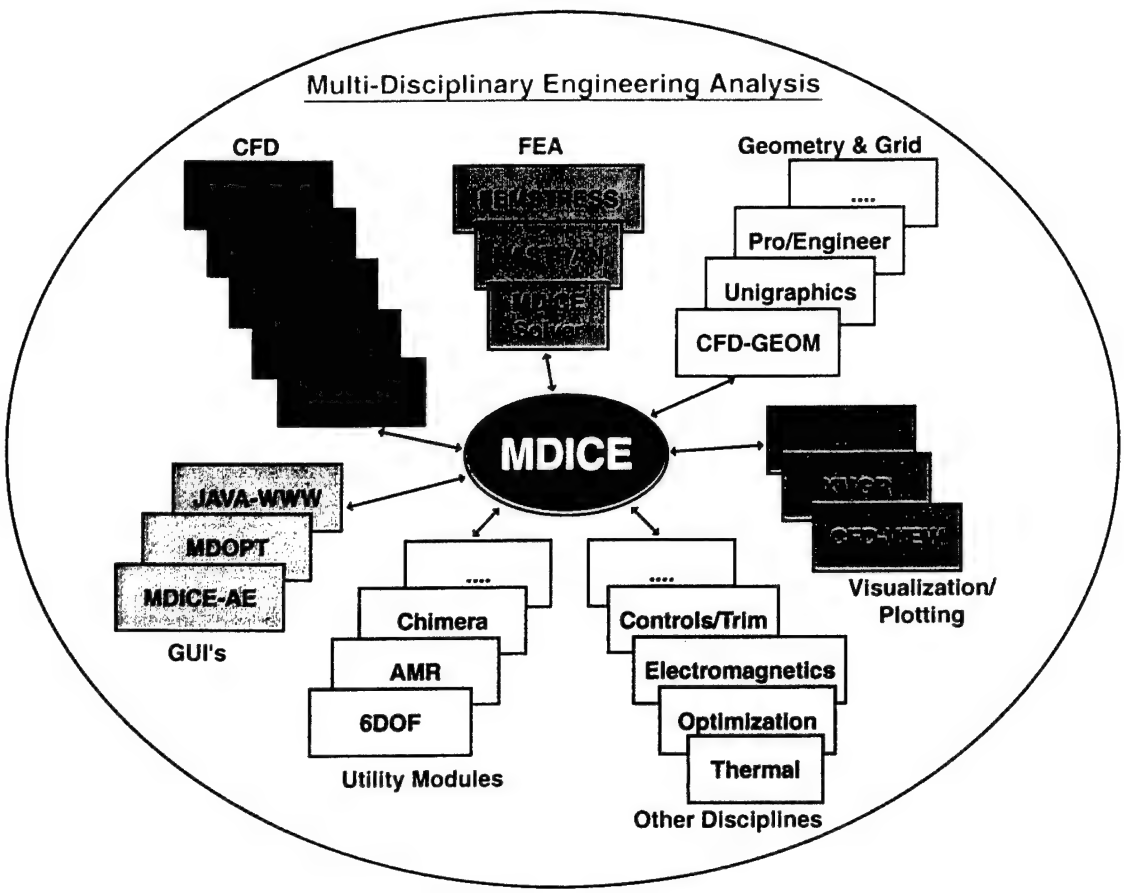


Figure 1. Conceptual Overview of the Multi-Disciplinary Computing Environment MDICE

The various engineering analysis codes and modules are under direct control by MDICE. From an MDICE script, those modules can be started and stopped by the MDICE user. Those modules can exchange data before, after, and most importantly during a run (e.g. per (sub-) iteration or time step). In addition, MDICE provides support for:

- Application synchronization,
- Temporal synchronization (proper time stepping of transient solvers),
- Grid and data conversions,
- Physics based interpolations across domain and discipline interfaces,
- The alignment of the various CSD and CFD patches with each other (multi-patching),
- System-wide restarts, and
- Heterogeneous distributed computing.

The MDICE framework can be run with a variety of graphical user interface options. Therefore, there is a strict separation between the Graphical User Interface (i.e. client) and the actual MDICE controller. The following options are available (see Figure 2):

- **No Graphical User Interface.** The MDICE controller is started from the command line. A script and simulation file name are command line arguments to the controller. This mode is ideally suitable for batch queuing systems.
- **Full Graphical User Interface.** This GUI allows full access to all MDICE's capabilities including the script editor. The current GUI is written in Motif, and runs

under UNIX only. In the near future (CY00), a FOX based GUI will be provided which can run natively on UNIX and Windows.

- **JAVA based Graphical User Interface.** The JAVA based interface can be run from any WWW browser. It presents the user with a list of all actively running MDICE simulations. By clicking on a simulation (hyperlink), the user can log into that simulation and trace progress for that particular simulation.
- **Special-Purpose Graphical User Interfaces.** The MDICE user can create their own interface for their particular application. MDICE-AE Graphical User Interface (written in FOX) is a good example. Another good example is the MDOPT GUI written in TCL/TK by Boeing.

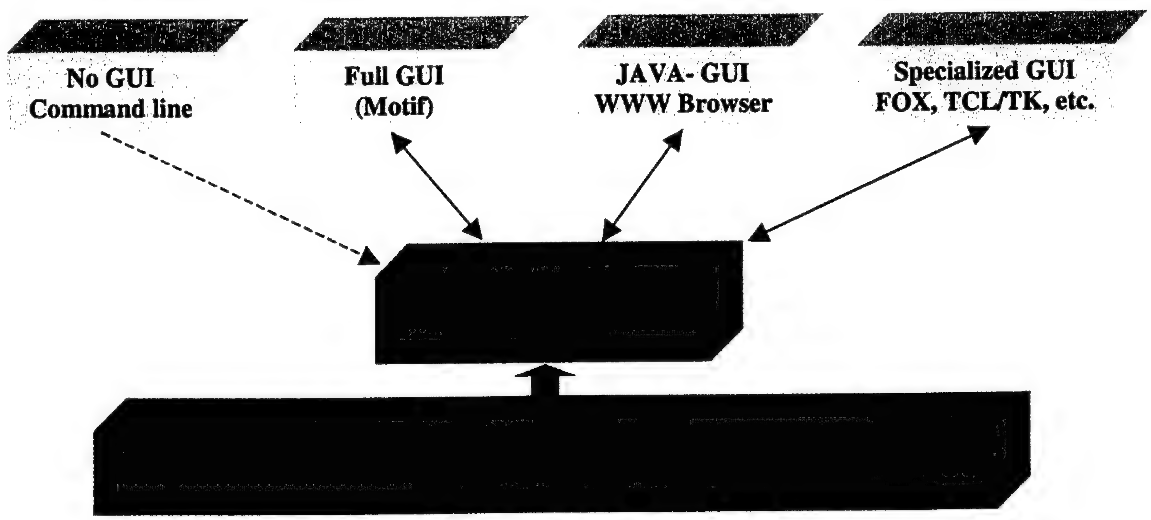


Figure 2. GUI Options for MDICE

In future versions of the MDICE code, the interface between the GUI and the MDICE Controller will be further standardized and fully programmable by the end-user. Besides graphical user interfaces, CFDRC envisions that one MDICE controller can act as a client to another MDICE controller. This means that one MDICE process can potentially control a large number of MDICE based simulations.

Figure 3 shows a more detailed conceptual overview of MDICE applied to aeroelastic applications (MDICE-AE).

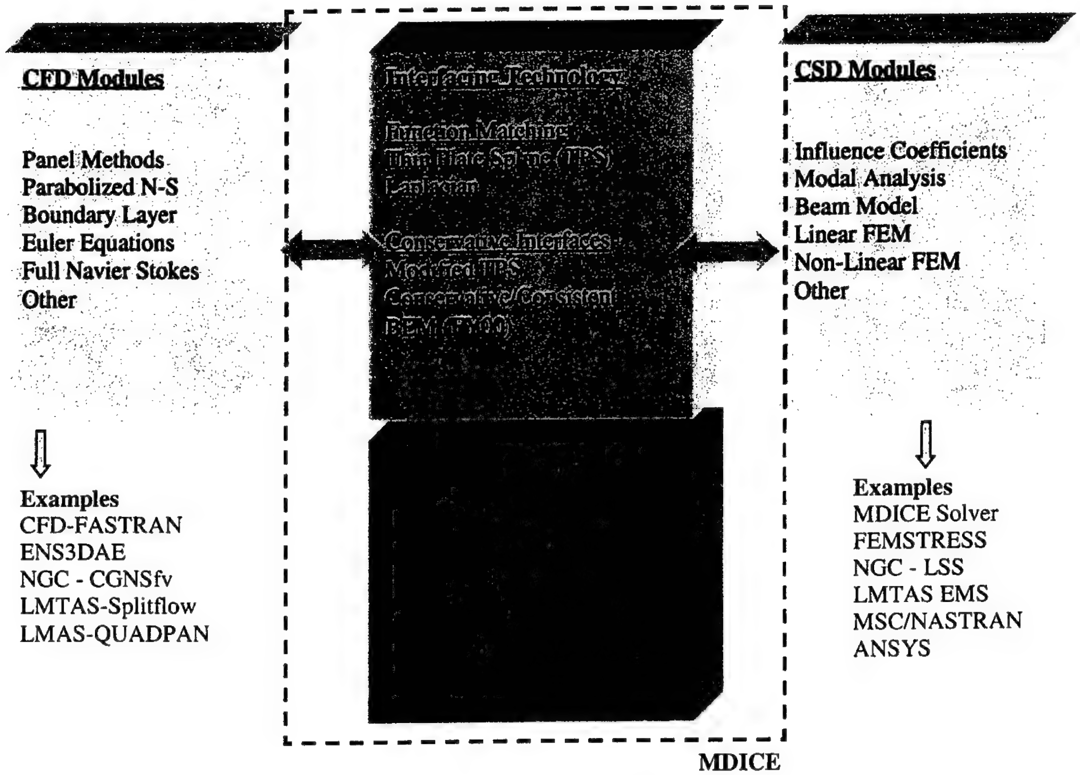


Figure 3. Conceptual Overview of MDICE-AE

Specifically for aeroelastic applications, MDICE provides an array of interfacing methods. Some interpolation methods are merely mathematical interpolation methods while others are more physics based and try to conserve 'virtual work' on either side of the interface. The Thin Plate Spline method, adopted from FASIT [1], is commonly used in aeroelastic codes. For this project, we have implemented, tested, and validated the consistent and conservative method by Brown [2]. Another very promising fluid-structure interface method based on the Boundary Element Method (BEM) by Zona Technologies will be implemented in MDICE in the FY00 time frame (as part of a different contract). The end-user can select any of those aforementioned interpolation methods by means of

the MDICE script. Currently the default method is the conservative/consistent method by Brown. A further investigation on the relative merit of the various fluid-structure interfacing methods is recommended, and is mentioned under the section "Recommendations for Future Work".

During model setup, the end-user labels all aeroelastic patches (boundary condition) with a number. This is done both in the CFD model and the CSD model. The MDICE system will automatically correlate the patches in the CFD and CSD model, i.e., patches with the same number constitute one fluid-structure interface. Each aeroelastic simulation may have one or more of these interfaces. The alignment of patches is automatic and uses a nearest neighbor search algorithm. All underlying details, e.g., structured vs. unstructured grids, grid or axis or orientation, cell centered vs. vortex based data, and metric conversions, are handled automatically as well. Besides having more than one fluid — structure interface, the aeroelastic simulation may have more than one structural solver (e.g., twin tail buffet simulation). Theoretically it is possible to combine various structured models in one simulation, e.g., modal analyses for wing and a beam model for the tail.

The term 'multi-patching' has been introduced to identify cases in which <u>one</u> fluid-structure interface is defined by multiple patches on the CFD and/or CSD side. This may happen when the flow solver uses a domain-decomposition scheme for parallel applications or when there is a detailed structural model with each component modeled separately, e.g., wing with models for leading edge flap, flaperon, inboard and outboard

wing box, etc. The multi-patch interface needs to be globally conservative. This means that all contributions to this patch need to be collected by the process which provides the largest contribution to this patch (in terms of number of grid points). Then the fluid structure interaction is performed as if it were one global patch. On the other side of the interface, the information is properly divided among all patches that make up that side. One has to keep in mind that each contributing patch on a multi-patch interface could physically reside in a different process and computer on the network.

The MDICE support for one or more multi-patching fluid-structure interfaces combined with the ability to run several structural solvers are critical for performing aeroelastic analyses on complex aircraft configurations.

4. PROJECT OVERVIEW

In the following sections a detailed overview of the various aspects of this project will be given. The first section outlines the developments for the Multi-Disciplinary Computing Environment MDICE. The next section details some of the engineering analysis applications that have been integrated with MDICE. In addition, a brief overview of some of the MDICE specific modules, which have been developed under this contract, is given. And finally, and overview of all software testing and validation activities is given.

4.1 Development of MDICE Computational Environment

This project has resulted in the development of the Multi-Disciplinary Computing Environment MDICE. MDICE enables engineering analysis codes to perform coupled multi-disciplinary analysis in a distributed computing environment. MDICE consists of a GUI, software libraries, API, and generic controller process to enable very dissimilar legacy analysis codes to dynamically exchange data with each other (e.g. provide grid, data and unit conversions, and facilitate arbitrary grid fluid-fluid interfaces and conservative/ consistent fluid-structure interfaces). The engineering analysis and design codes may be from any source, i.e. CFDRC, proprietary, third party software vendor, US-Government, or public domain. The engineering disciplines supported include: parametric CAD, grid generation, computational fluid dynamics, structural analysis, heat transfer, controls, visualization/animation/ plotting, optimization, etc. More disciplines and engineering analysis codes may be added on an as needed basis.

The development and testing of the MDICE parallel distributed computing environment have been a major part of this contract. It was based on the Visual Computing Environment VCE (as sponsored by NASA Glenn). The following achievements have been made.

- Design, development and testing of all objects which describe the data in MDICE. Objects have been created for structured grids, unstructured grids, polyhedral grids, data, fluid-fluid interface object, fluid-structure interface object, point data object (used for monitoring points in any code), and so on.

- Data conversion needs to be done while the data are being exchanged, therefore MDICE has extensive support for many grid and data conversions, such as:

  - Left handed vs. right handed grids,
  - Single to double precision, and vice versa,
  - 0-based to 1-based arrays, and vice versa,
  - Generic support for unit conversions (e.g. British to metric system)
  - Conversion for dimensionalized to non-dimensionalized data, and vice-versa.

  An important feature is that any of those conversions can be combined (if appropriate). In addition, the data duplication is kept to a minimum (usually none).

- In particular, fluid-structure interface technology has been developed. Several interfaces are available in MDICE, such as the Conservative/Consistent interface by Brown [2], and the Thin Plate Spline method from the FASIT software [1]. The interface assembly, i.e. determining which CFD patch 'talks' to which CSD patch, is fully automated based on boundary condition labels specified during the model set-up phase. Low level issues such as patch orientation, cell-centered versus vertex based data, structured versus unstructured grids are handled automatically by the MDICE software.

- An API has been developed and documented for creating and manipulating the MDICE objects from an application program. This has been implemented in the form of a library which contains approximately 100 functions. Those functions can be called from any C, C++, F77, or F90 program. All platform specific marshalling of the function parameters is taken care of in MDICE.

- With the script the end user controls what data and when it needs to be exchanged, and between which codes. Many improvements to the script have been made, including:

  - Timing of script operations,
  - Break points in script and line-by-line execution of script,
  - Parallel execution of script segments,
  - Unix Shell commands are supported from the script,

- Start and stop of modules,
- Support for pseudo time stepping, and
- Automated interface assembly.

- Porting to all Unix and PC (Windows) platforms. The software runs on SGI (version 5.X, 6.X), DEC, HP (including the parallel Examplar), SUN, IBM, PC (NT, 98, Linux). The software has been ported and tested on SGI supercomputers under control of a batch queuing system such as PBS.

- All MDICE modules run in parallel with each other, however certain engineering analysis codes may have parallel versions of their own. This feature is explicitly supported in MDICE and is called multi-level parallelism.

- MDICE may have several graphical user interfaces. In future versions of MDICE there will be a clear distinction between the MDICE controller and the MDICE graphical user interface, i.e. they are separate executables. The interfaces between the two software pieces will be documented and application specific MDICE GUI's can be created by the end-user. An example of a user developed GUI is the MDOPT GUI by Boeing. CFDRC has several GUI's for MDICE, the main GUI gives access to all MDICE features and should be used when the most flexibility is required. The MDICE-AE GUI is a specialized graphical user interface for performing fluid-structure interaction simulations. An optional JAVA based client interface has been developed as well. With a web browser, the user can see how many MDICE

simulations are running and where. When the user clicks on one simulation (hypertext link), the user gets access to one specific MDICE controller with all details of that particular simulation.

4.2 Integration of Engineering Analysis Codes

The integration of programs into MDICE can be accomplished by a direct coupling, i.e. invoke MDICE API from program, or by using a wrapper which encapsulates the standard file I/O of a program. A direct interface is preferred, but sometimes the wrapper approach is necessary if source code is not available. A standard example of a wrapper-based interface is delivered with MDICE. In addition, some simple examples of directly integrated programs are delivered with MDICE. For more complex examples such as entire flow solvers please contact AFRL or CFD Research Corporation. The time it takes to integrate a complex application program is in the order of one week.

CFD Research has taken an open approach to developing and maturing MDICE. During the course of this project we have collaborated with many organizations which could contribute to the MDICE developments, testing, and validation. One of the results is that a variety of engineering analysis programs has been integrated into the MDICE environment. Certain codes have been integrated by CFD Research Corporation while others have been integrated in collaboration with other organizations, and/or by those other organizations themselves. Another important benefit is that while we were developing MDICE we have had a lot of feedback from companies such as Northrop

Grumman and Lockheed Martin. The fairly large number of integrated engineering analysis codes clearly demonstrates that MDICE has sufficient generality and flexibility to handle a large number codes with all their differences and peculiarities.

As of the writing of this report, the following codes have been integrated with MDICE.

Organization	**Code**	**Description**
CFDRC	CFD-GEOM	Geometry Modeling and Grid Generation
	CFD-VIEW	Data Visualization and Animation
	CFD-ACE	General Purpose Flow Solver (structured grids)
	CFD-ACE+	General Purpose Flow Solver (polyhedral grids)
	CFD-FASTRAN	External Aerodynamics Flow Solver
	MDICE-Solver	Structural Solver (FEM, Modal Analysis, Beam Models, I.C.)
	FEM-STRESS	Structural Analysis Code
	CFD-FastBEM	Boundary Element Method Solver
	PRO/Engineer	Parametric CAD (Parametric Technology Corp.)
	Unigraphics	Parametric CAD (Unigraphics Solutions)
	MSC/NASTRAN	Structural Analysis (File Based Interface, MSC Software)
	Ansys	Structural Analysis (File Based Interface)

	XMGR	Line Plotting Tool (Public Domain)
	ImageMagic	Visualization/Animation Tool (Public Domain)
AFRL	COBALT_60	External Aerodynamics (polyhedral grids)
	ENS3DAE	Aeroelastic Analysis Code
	CAP-TSD	Flow Solver (To be implemented FY2000)
Northrop	GCNSfv	Parallel Flow Solver
	LSS	Linear Structures Solver
LMTAS & GE	Splitflow	External Aerodynamics Flow Solver (polyhedral grids)
	EMS	Influence Coefficient Solver
	Ansys	Structural Analysis Code
	MSC/NASTRAN	Structural Analysis Code
Boeing	MDOPT	Multi-Disciplinary Optimization Code
LMAS & Georgia Tech Univ	Quadpan	Linear Flow Solver (Panel Method, Under Development)

NASA GRC	ADPAC	Parallel Flow Solver for Compressors
	NPARC	Parallel Flow Solver for Inlets
P&W	NISTAR	Flow Solver (Turbines)
	NASTAR	Flow Solver (Compressors)
	CORSAIR (NCC)	Flow Solver (Combustors, NCC - National Combustor Code)
Texas A&M	Abaqus	Structural Analysis (HKS Inc.)

Notes:

1. This is a comprehensive list. Not all programs have been integrated with MDICE as part of this project.
2. The rights of the various engineering analysis programs and the associated MDICE interfaces remain with the rightful owner (this includes US Government Codes).
3. The primary codes for fluid-structure interaction during this project were: CFD-FASTRAN (CFD), MDICE Solver (CSD), MSC/NASTRAN (CSD), CFD-VIEW (visualization), and XMGR (visualization).

4.3 Development of MDICE Modules

The focus of the contract was to reuse existing engineering analysis modules and to integrate the various software programs by means of MDICE. However, one module has

been specifically developed to work in conjunction with the MDICE software. This is the MDICE Structural Interface Module.

4.3.1 MDICE Structural Interface Module (Structural Solver)

A variety of methods for characterization of the structural properties of an aircraft structure are being used for aeroelastic analysis by the aerospace community. Most commonly used are influence coefficients, modal analysis, linear FEM, and a variety of specialized beam models. To facilitate all those methods the MDICE Structural Interface Module has been developed. One or more of those modules can be part of one aeroelastic simulation. The module has solvers for Influence Coefficients, FEM, Modal Analysis, and Beam models. Both steady state and transient simulations are supported. The actual models, i.e. mode shapes, mass and stiffness matrices, etc., need to be defined elsewhere (e.g. MSC/NASTRAN). Optionally, for steady state simulations the MSC/NASTRAN solver can be invoked from this module. The Structural Interface Module will prepare a proper input file for MSC/NASTRAN, launch MSC/NASTRAN, and will read the results (deflections) back from the NASTRAN file and make it available to MDICE.

A graphical user interface has been written to help the user in setting up an aeroelastic simulation. Figure 4 shows two panels from the Structural Interface Module.

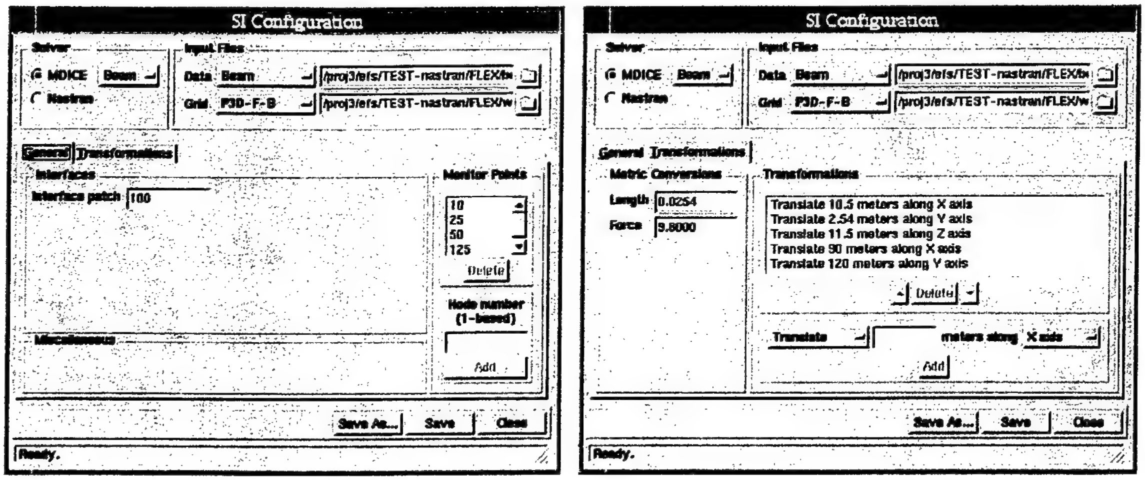


Figure 4. Two Examples of the MDICE Structural Solver GUI

The objective is that the user does not have to modify any of the input files to the structural solver. By means of this graphical user interface, the user can specify which solver needs to be run (MDICE solver, or MSC/NASTRAN), the appropriate input file names, fluid-structure interface patch numbers, some optional monitor points for plotting applications, some metric conversions, and some transformation for proper alignment of the CFD and the CSD model. More detailed information can be found in the MDICE-AE User's Manual.

4.4 Demonstration and Validation Cases

The MDICE computing environment has been used to demonstrate and validate various aeroelastic simulations. The default codes that have been used to perform those simulations are CFD-FASTRAN for CFD, the MDICE Solver or MSC/NASTRAN for structural analysis, CFD-VIEW for visualization, and XMGR for line plotting. Those simulations included the steady state AGARD 445.6 deflection analysis, the transient

AGARD flutter analysis, the steady state F16-like (VAT) wing body configuration, the single tail buffet analysis on generalized aircraft, and the twin tail buffet study on generalized aircraft. All those studies have been validated with experimental data.

More detailed descriptions of the results can be found in the various papers mentioned in the references for each section. It needs to be mentioned that the twin tail buffet validation study has been followed up by an SBIR Phase I contract for passive and active flow control on this configuration. Some early results are described in several papers mentioned in the references. In Phase II, the methodology will be applied to the F18C full aircraft.

Movies (in animated GIF or AVI format) are available from CFDRC showing the convergence of the AGARD deflection history and the transient twin tail buffet simulation results.

4.4.1 AGARD 445.6 Wing

Introduction

In this chapter, the various results for the AGARD 445.6 Wing are presented. A variety of cases have been run, i.e. Euler and Full Navier Stokes, and Conservative/Consistent interpolation versus 'interpolated mode shapes'. The 'interpolated mode shapes' refers to a practice of taking the mode shapes and interpolating them to the CFD grid and reducing the general fluid-structure interface problem to a 1-to-1 grid connectivity problem on the Outer Mold Line (OML).

Simulation Characteristics

The following parameters were used for this problem:

- Static Analysis: M = 0.8; AOA = 1.0 deg
- Transient Analysis: M = 0.96, AOA = 0.0 deg
- First 14 mode shapes were used
- For flutter calculation: impulse in first mode velocity at time zero
- 99,840 nodes on CFD mesh, 231 nodes on structural mesh
- Time step of 1e-4 sec (2000 time steps for flutter simulation)
- Three subiterations were used for fluid-structure coupling
- CSD: SI, Modal Analysis, or FEM (MSC/NASTRAN)
- CFD: Roe's scheme for spatial differencing, first order temporal differencing (flow solver: CFD-FASTRAN)

Geometry

Figure 5 shows the CFD mesh and the wing.

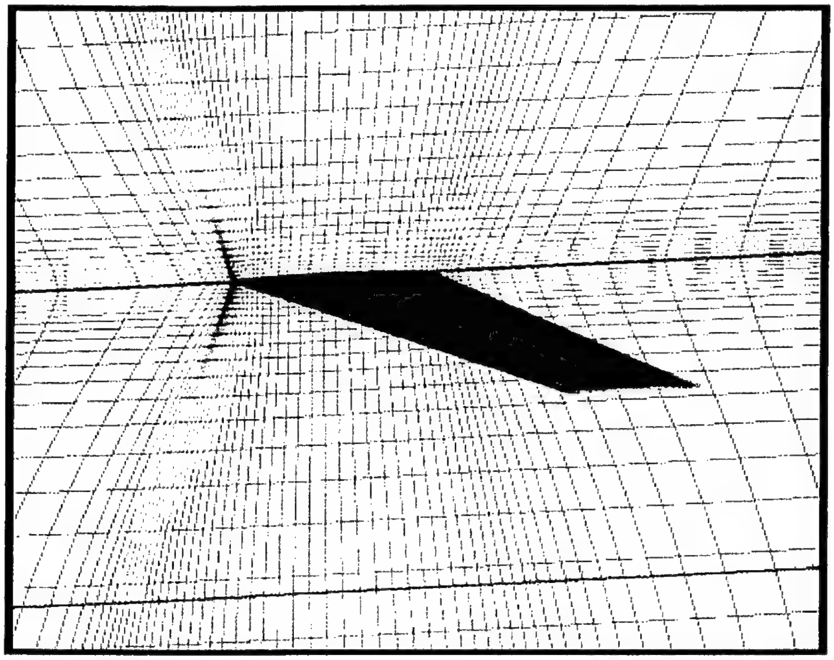


Figure 5. Computational CFD Mesh for AGARD Wing

Results

The following sections describe the various results for this case. This case has been run with CFD-FASTRAN as the flow solver and structural interface (SI, modal analysis) as the structural solver. In addition, several cases have been run with CFD-FASTRAN and MSC/NASTRAN (corresponding FE model) and nearly identical results were obtained.

Steady State

The following two images, Figure 6, show the results of the steady state aeroelastic simulation. The picture on the left side shows the displacement as a function of the number of iterations between the fluids and the structures codes. Full aeroelastic

convergence was reached within 25 exchanges between the fluid and the structures codes. The leading tip deflection was 1.12 cm, while the trailing tip deflection was 0.98 cm. On the right side, the actual data on the fluid-structure interface is visualized. The left side (reflected wing) shows the Pressure, while the right side shows deflection (including deflection vectors).

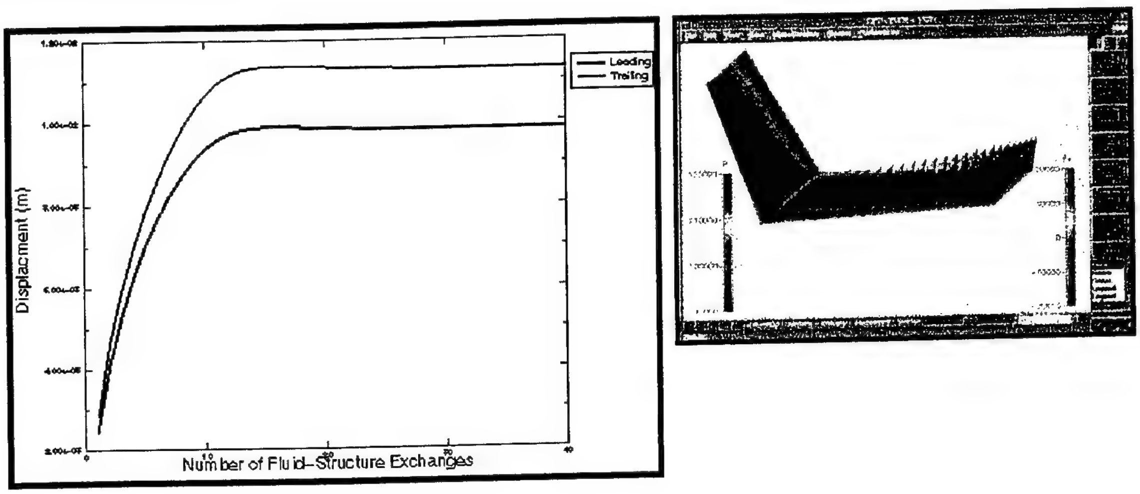


Figure 6. Steady State Solution with Leading and Trailing Edge Displacement

Original vs. Interpolated Mode Shapes

The original mode shapes refer to the modes on the original FEM mesh (with MDICE consistent/conservative interpolation), while the interpolated mode shapes have been interpolated from the FEM mesh to the CFD mesh, Figure 7. For the steady state case, the generalized displacements of both simulations were virtually identical for the first mode, the interpolated generalized displacements was 0.4% greater than that obtained by using the original mode shapes.

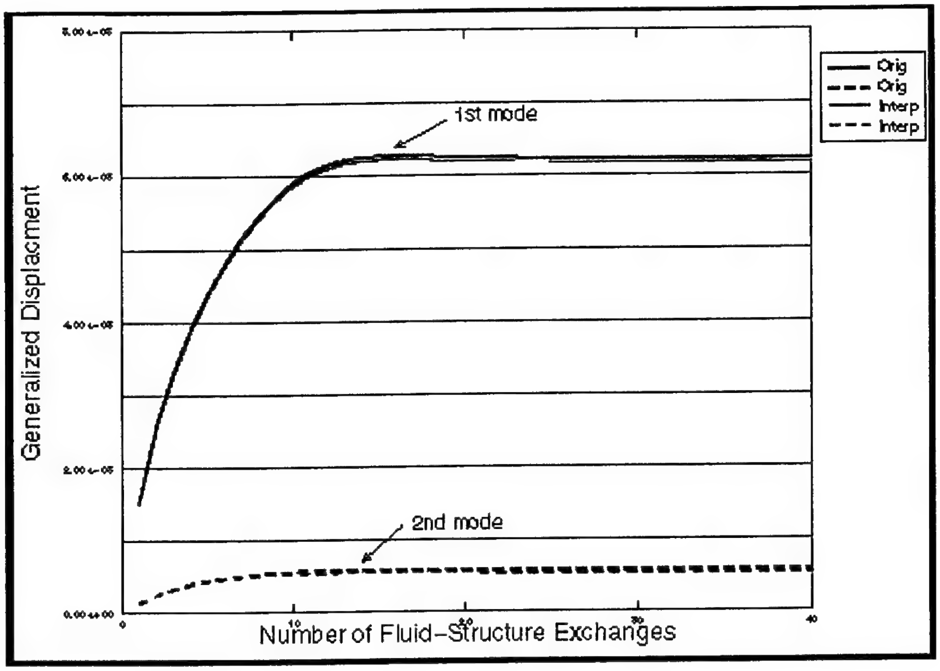


Figure 7. Original vs. Interpolated Mode Shapes

Transient Flutter (Inviscid)

This simulation was run using original modes and the MDICE conservative/consistent interface, Figure 8. The time step was 1e-3 sec, and the q/qflutter = 0.8, 1.0, 1.2. Increasing q/qflutter from 0.8 to 1.2 yielded an increasing amplification factor and increasing frequency (A= 0.977, 1.011, 1.064 and freq = 73.92, 76.62, and 80.55 rad/sec for q/qflutter of 0.8, 1.0, 1.2, respectively). 'qflutter' is the dynamic pressure at the experimental flutter point. For this case (1e-3 sec. time step) a computational time of 2 hours (on DEC workstation cluster - 3 CPU's). For the 1e-4 sec. time step case the computational time increases to 20 hours.

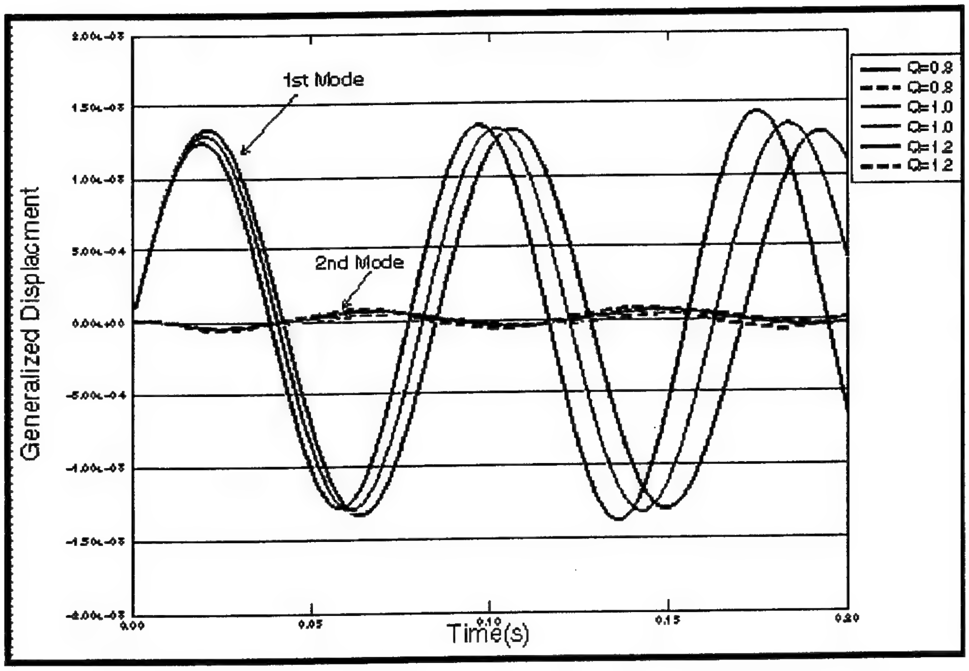


Figure 8. First Two Mode Shapes (Flutter)

Transient Flutter (Original vs. Interpolated Mode Shapes)

The usage of interpolated mode shapes resulted in an increased amplification factor and an increased frequency (A = 1.05, {original = 1.01}, omega = 79.5 rad/s {original = 76.6 rad/s}) as shown in Figure 9. Both simulations were run at 1e-3 time step and at q/qflutter = 1.0.

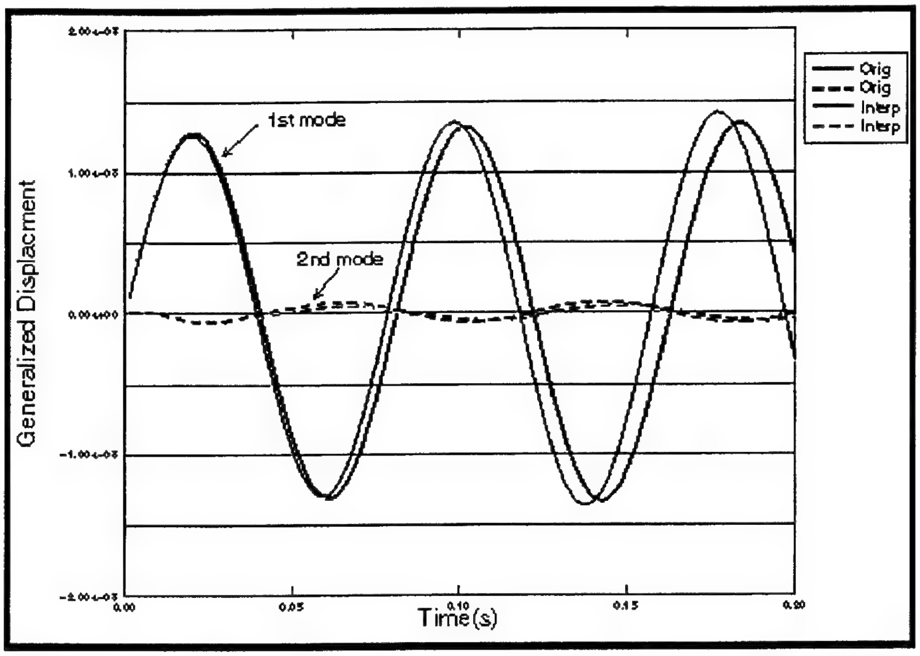


Figure 9. First Two Mode Shapes (Original vs. Interpolated Mode Shapes)

<u>Flutter Summary (Interpolated vs. Original Mode Shapes)</u>

The following graphs (Figure 10) show the amplitude and the frequency as a function of q/qflutter (which is 0.8, 1.0, and 1.2). The predicted flutter point for the original mode shapes was at q/qflutter = 0.92, while the predicted flutter point for the interpolated mode shapes was at q/qflutter = 0.81. Again, very clearly visible is that the interpolated mode shapes result in increased amplification factor and frequency.

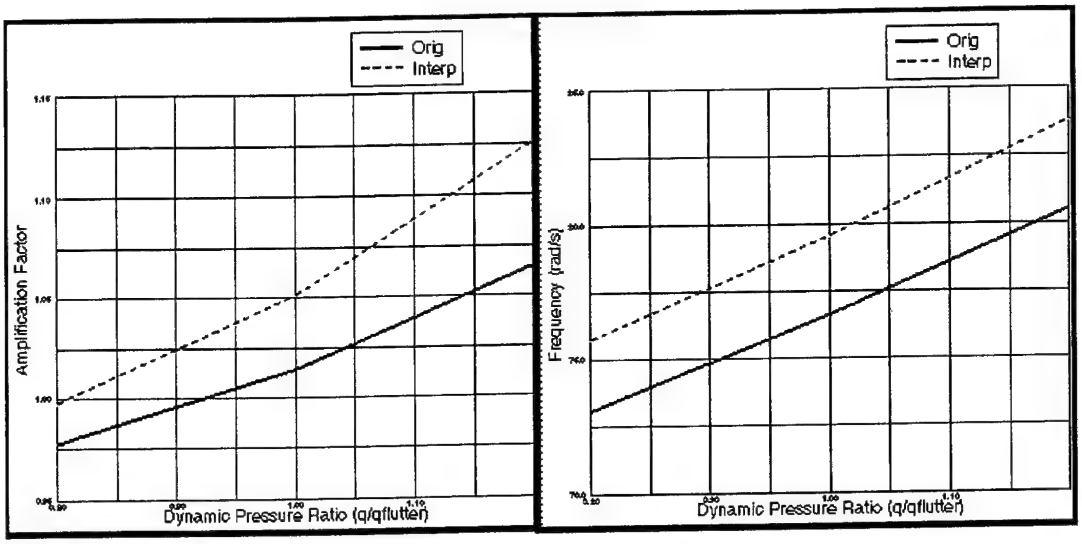


Figure 10. Overview of Original vs. Interpolated Mode Shapes for Flutter Calculations

Flutter (Viscous)

This simulation was run with original mode shapes and MDICE's consistent/conservative interpolation method. Time step is 1e-4. Increasing q/qflutter from 0.8 to 1.2 yielded an increasing amplification and an increasing frequency (A = 0.966, 0.986, and 1.014 and omega = 73.1, 76.6, and 80.2 rad/s for q/qflutter of 0.8, 1.0, and 1.2 respectively, Figure 11.

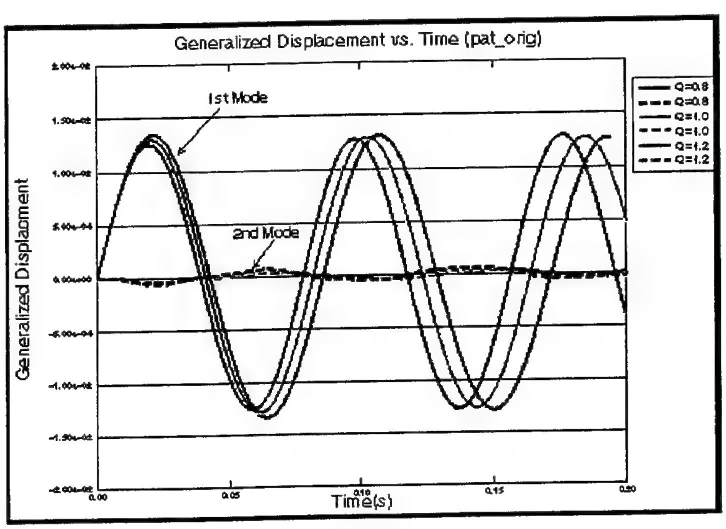


Figure 11. First Two Mode Shapes of Viscous Flutter Calculation

Flutter Comparison (Original vs. Interpolated Mode Shapes)

Time step is 1e-3, and the q/qflutter is 1.0. In this simulation the original and interpolated mode shapes are compared for the viscous simulation, Figure 12. The use of interpolated mode shapes resulted in an increased amplification factor and an increased frequency (A = 1.011 versus original is 0.986, and omega = 79.4 versus original is 76.6 rad/s).

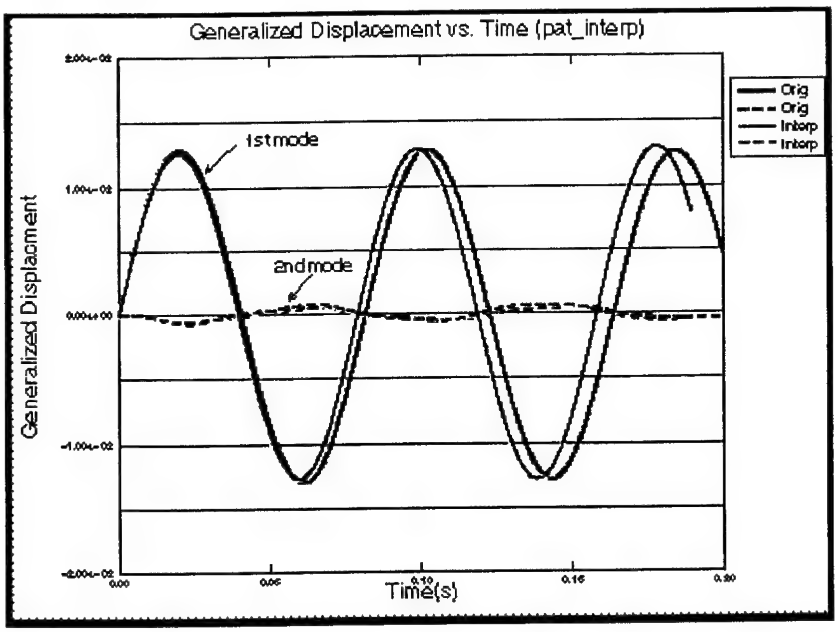


Figure 12. Original vs. Interpolated Mode Shapes for Viscous Flutter Simulation

Flutter Comparison (Viscous, Roe vs. Van Leer)

The use of a spatial discretization scheme with increased dissipation resulted in a damped response for a previously undamped case. Baseline case is Roe's Approximate Riemann solver (A = 1.05) versus Van Leer's Flux Vector Splitting (A = 0.971), Figure 13.

Interpolated mode shapes, q/qflutter = 1.0.

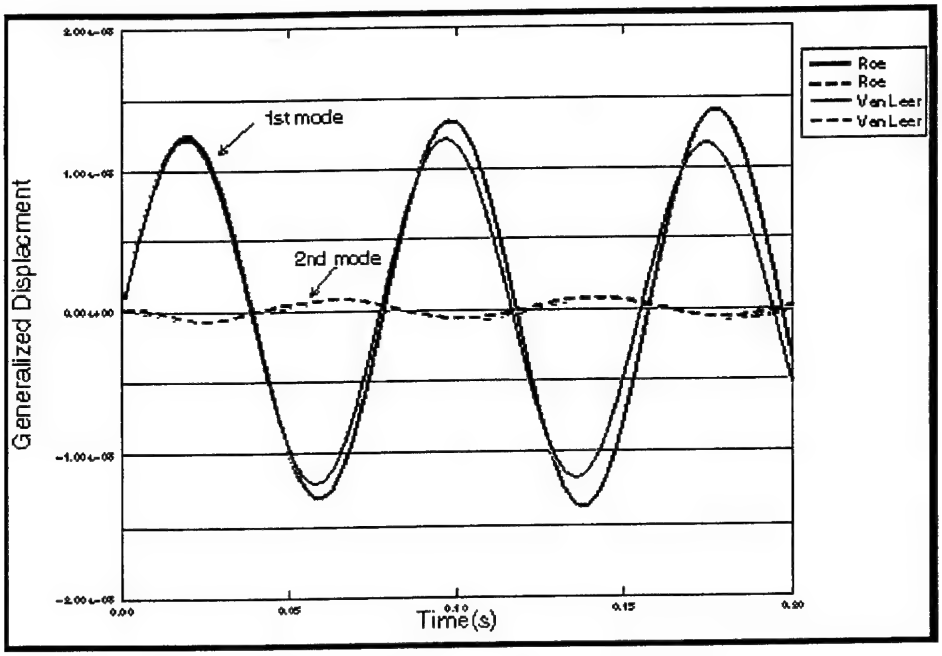


Figure 13. Roe vs. Van Leer Sensitivity on Flutter Simulation

<u>Flutter - Time Step Sensitivity</u>

The use of a large time step (1e-3 sec) for the viscous simulation resulted in a highly amplified response, whereas a time step of 1e-4 sec yielded a slight damped response, Figure 14. One more simulation is required here (with time step 1e-5 or 5e-4).

Original modes, consistent/conservative interpolation, q/qflutter = 1.0.

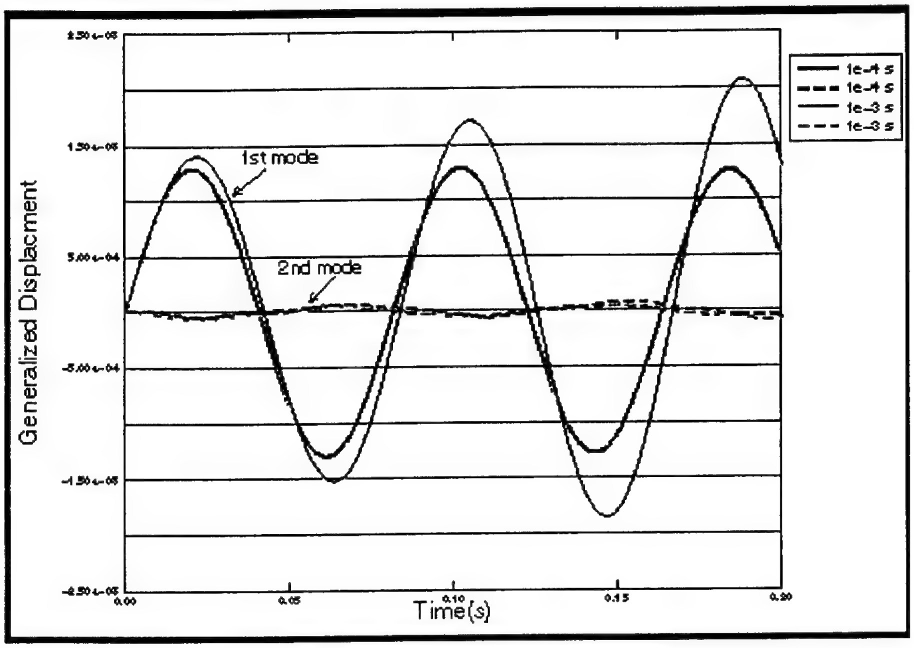


Figure 14. Time Step Sensitivity on Flutter Simulation

Conclusions

In summary, the following conclusions can be made from this study.

- There is no significant difference between interpolated and original modes in the steady analysis of the AGARD wing at 1.0 degree AOA.
- Both the inviscid and viscous transient flutter analysis provided accurate predictions of the flutter point
- The interpolated mode shapes yielded higher frequencies and amplification factors for the transient flutter analysis
- The flutter analysis has a high level of sensitivity to numerical parameters, this should be analyzed in more detail in future studies.

4.4.2 F16-like VAT Wing Body Analysis

Introduction

A steady state aeroelastic simulation was performed on the F16-like wing body configuration. Influence coefficients were used on the structures side (MDICE-SI solver), while a FNS simulation was run on the CFD side (CFD-FASTRAN flow solver).

Simulation Characteristics

The following parameters were used for this simulation:

- Flight conditions: AOA = 5.116 degrees, M = 1.2, CFL = 10, T = 245.68 K, Rho = 0.505 kg/m^3, and P = 3.56 x 10^4 N/m^2.
- Euler flow assumption.
- CFD: Roe's scheme for spatial differencing, first order temporal differencing (flow solver: CFD-FASTRAN).
- CSD: SI, Influence coefficient model.
- An under relaxation factor in the grid deformation was used to prevent grid overlapping due to excessive deflections.

Geometry

The model consists of a flexible wing and rigid wing strake and fuselage. Figure 15 shows the surface grid of the configuration model.

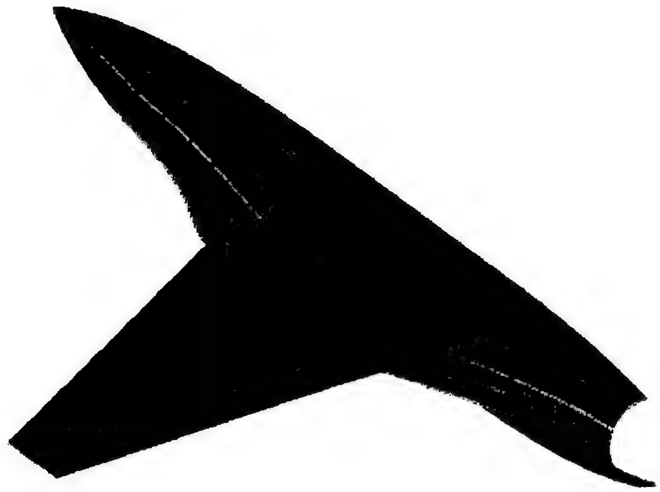


Figure 15. Computational CFD Grid

Results

Some of the aeroelastic results of this case are shown here. The problem is solved first for the initial conditions using rigid configuration. Next, the problem is solved with flexible wing starting from the initial conditions obtained in the first step.

Figure 16 shows the surface pressure contours over the deflected configuration model. The final wing-tip section displacement was computed of 65 millimeter. The experimental displacement for this case was 68 millimeter.

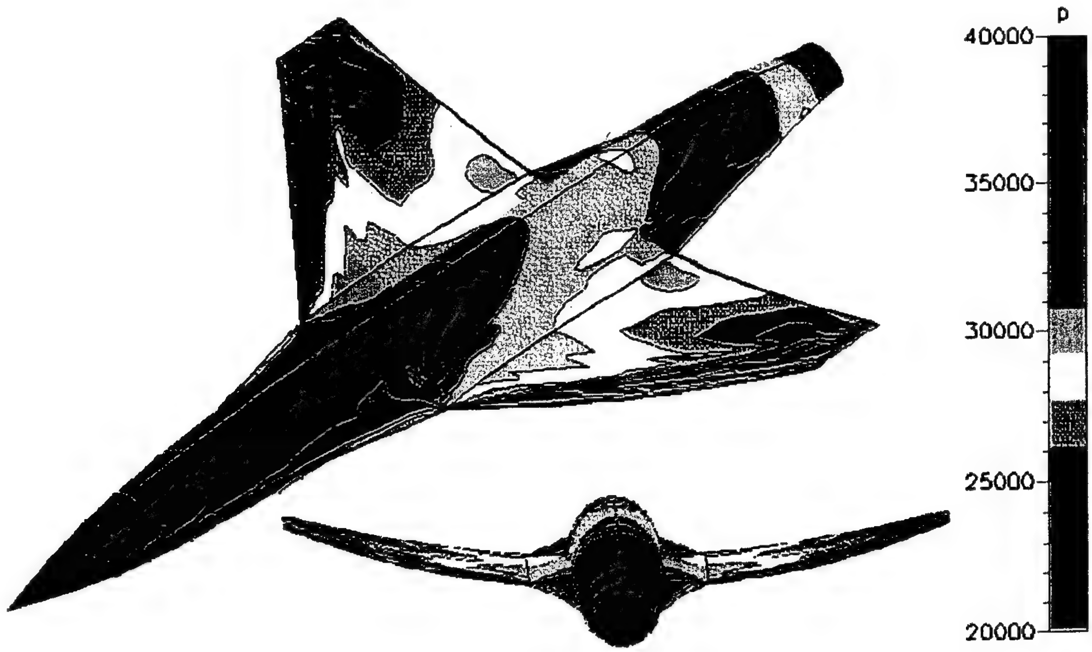


Figure 16. Surface Pressure Contours on Deflected Geometry

Figure 17 shows the wing tip displacement and the wing midspan displacement as a function of the number of fluid-structural exchanges. Full convergence was achieved within 50 fluid-structure exchanges.

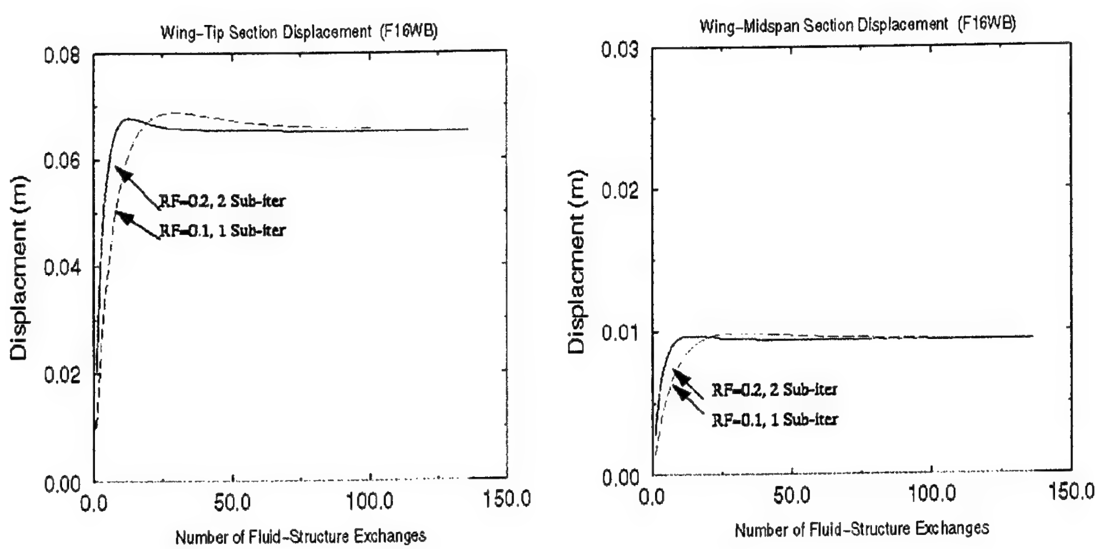


Figure 17. Wing Tip and Midspan Displacement as a Function of the Number of Fluid-Structure Exchanges

4.4.3 Single Tail Buffet Analysis

Introduction

In this section, the problem of single-tail buffet of a generic fighter aircraft is simulated and presented. The tail is clamped at the root and is oriented normal to the delta-wing surface. The vertical tail was modeled as a cantilevered beam fixed at the root. The tail was allowed to oscillate in both bending and torsion modes.

Simulation Characteristics

The following parameters were used for this simulation:

Flight conditions: M = 0.4, Re = 1.25 x 10^6, AOA = 36 deg.

- First 6 bending modes and first 6 torsion modes were used.
- The CFD grid is a multi-block H-H grid structure consisting of 8 blocks. The total size of the grid is 215,000 grid points.
- The CSD grid is a one-dimensional grid of 125 grid points.
- Time step of 1e-4 sec.
- CFD: Roe's scheme for spatial differencing, first order temporal differencing (flow solver: CFD-FASTRAN).
- CSD: SI, Beam model analysis.

Geometry

The configuration model consists of a 76^{o}-swept back, sharp-edged delta wing of aspect ratio of one and a flexible, rectangular tail of aspect ratio of 1 placed at the geometric

symmetry plane. The vertical tail is oriented normal to the upper surface of the delta wing. Figure 18 shows a surface grid of the delta-wing/single-tail configuration model. The next figure shows a portion of the three-dimensional grid.

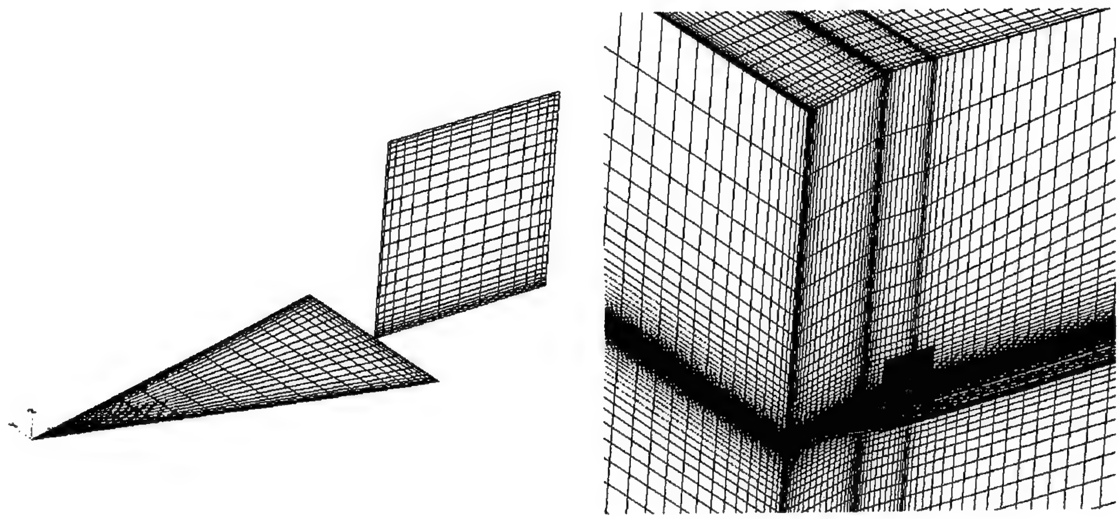


Figure 18. Computational CFD Grid

Results

The following sections describe some of the results of this case. This case has been run with CFD-FASTRAN as the flow solver and SI (Beam modal analysis) as the structural solver. First, the single tail is assumed rigid to obtain the initial conditions of the aeroelastic simulation. Next, the flexibility of the tail is turned on and solution is advanced starting from the initial conditions obtained in the first step.

Figure 19 shows a three-dimensional view, side view and front view of the iso-total-pressure surfaces over the rigid configuration model. The figure shows the asymmetric vortex-breakdown of the leading-edge vortices ahead of the vertical tail.

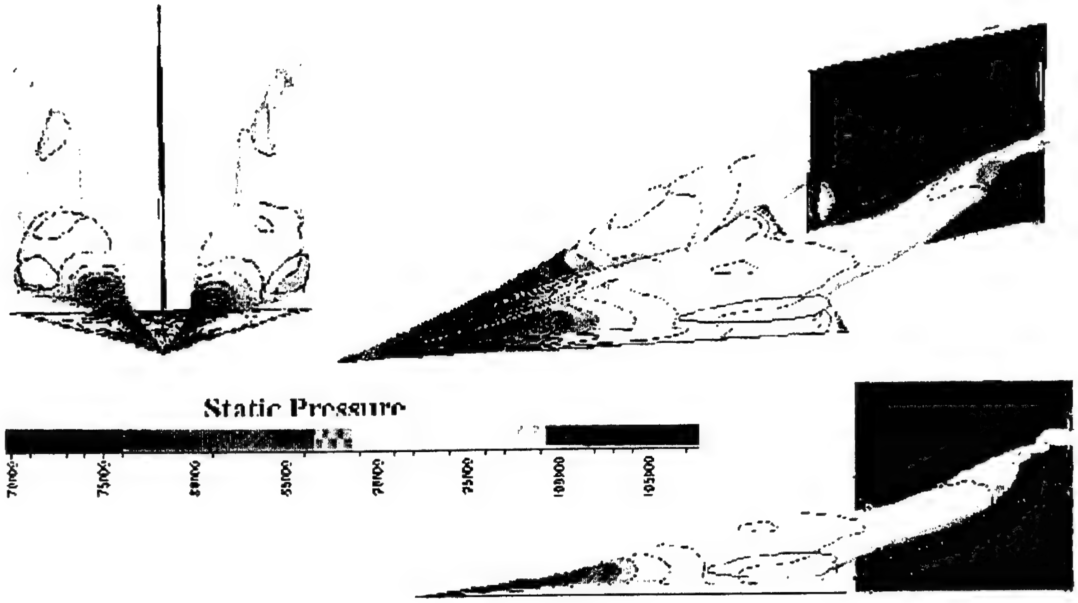


Figure 19. Iso Total Pressure Surfaces for Rigid Model

Figure 20 shows the same views for the flexible tail case. The figure clearly shows the substantial effect of the tail flexibility. The vortex breakdown becomes stronger and moves upstream due to the motion of the tail. The motion of the tail causes the vortical flow to change its path.

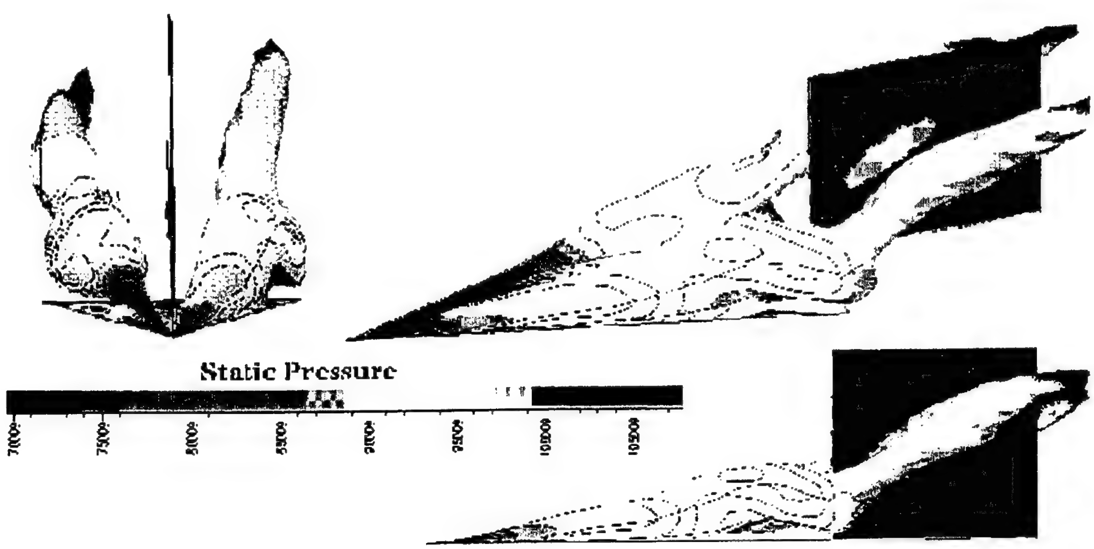
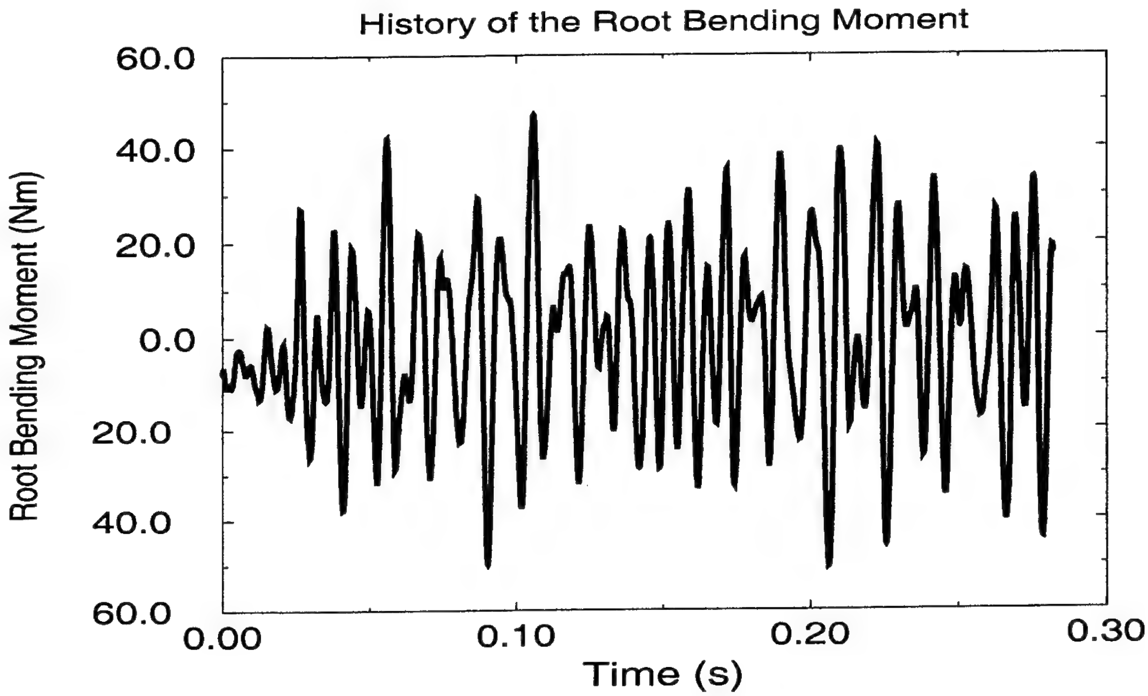


Figure 20. Iso Total pressure Surfaces for Flexible Model

Aeroelastic Results

Figure 21 shows the history of the tail-root bending moment and the histories of the tail-tip bending and torsion deflections. The frequency of the torsion deflection is almost twice that of the bending deflection.



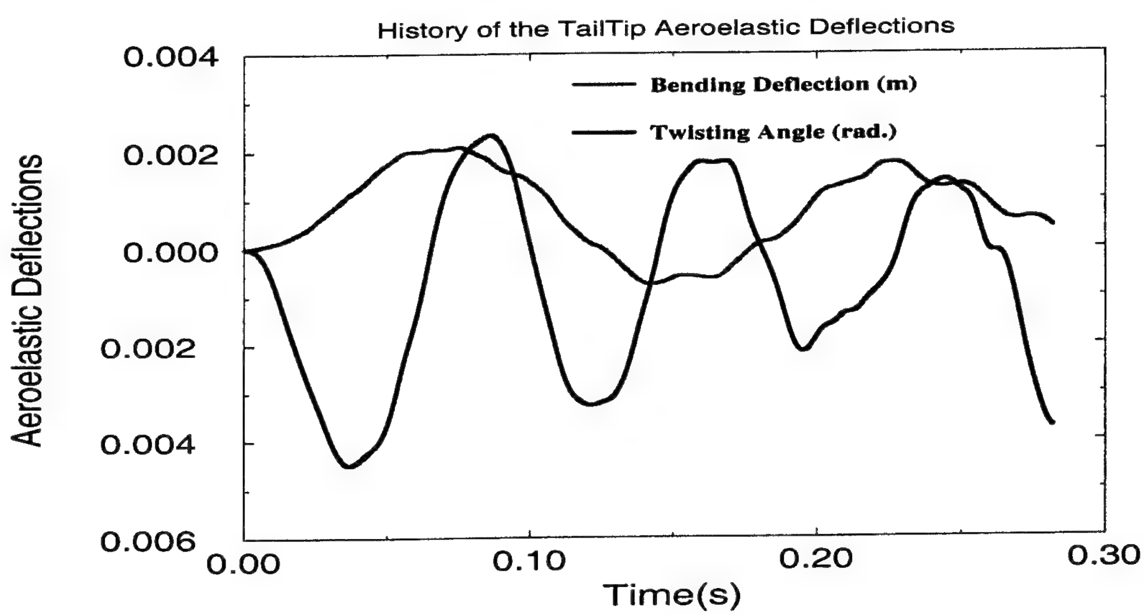


Figure 21. Root Bending Moment and Tail-Tip Bending and Torsion Deflections

Conclusions

In summary, the following conclusions can be made from this study.

- The tail flexibility has a substantial effect on the aeroelastic responses and the flow field.
- The frequency of the torsion deflections was almost twice that of the bending deflections.

4.4.4 Twin Tail Buffet Analysis

Introduction

In this section, the state-of-the-art problem of twin-tail buffet of generic fighter aircraft is simulated and presented. In a buffet condition, the leading-edge vortices of a delta wing break down before reaching the twin tails producing an unsteady turbulent flow which impinges on the surfaces of the tails, causing severe premature structural fatigue. The vertical tails were modeled as cantilevered beams fixed at the root. The tails were allowed to oscillate in both bending and torsion modes. The phenomenon is predicted and the computational results are validated against the experimental data of Washburn et al.[4].

Simulation Characteristics

The following parameters were used for this simulation:

- Flight conditions: M = 0.4, Re = 1.25 x 10^6, AOA = 20-40 deg.
- First 6 bending modes and first 6 torsion modes were used.

- The CFD grid is a multi-block H-H grid structure consisting of 20 blocks. The total size of the grid is 450,000 grid points.
- The CSD grid is a one-dimensional grid of 125 grid points.
- Time step of 1e-4 sec.
- CFD: Roe's scheme for spatial differencing, first order temporal differencing (flow solver: CFD-FASTRAN).
- CSD: SI, Beam model analysis.

Geometry

The configuration model consists of a 76°-swept back, sharp-edged delta wing of aspect ratio of one and dynamically scalled, flexible, swept back twin tail of aspect ratio of 1.4. The twin tails were shaped after Washburn et al.[4]. The vertical tails are oriented normal to the upper surface of the delta wing and have a leading edge sweep of 62.5 degrees. The separation distance between the twin tails is 78% of the wing span. Figure 22 shows a surface grid of the delta-wing/twin-tail configuration model.

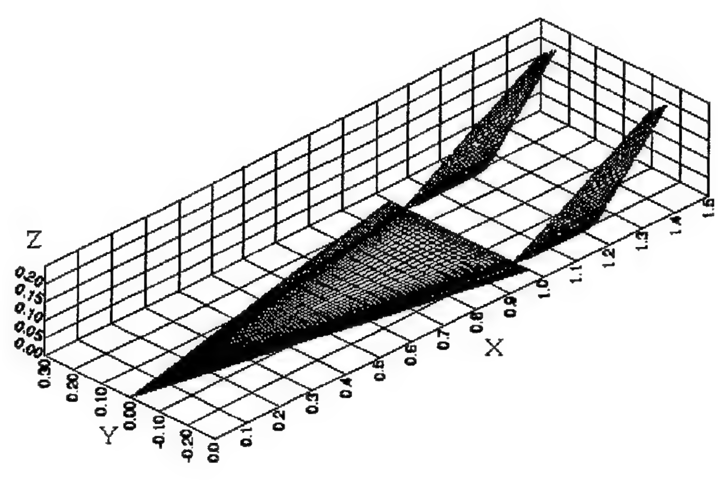


Figure 22. Twin Tail Configuration

Results

The following sections describe the various results for this case. This case has been run with CFD-FASTRAN as the flow solver and SI (Beam modal analysis) as the structural solver. First, the twin tails are assumed rigid to obtain the initial conditions of the aeroelastic simulation. Next, the flexibility of the tails are turned on and solution is advanced starting from the initial conditions obtained in the first step.

The following two figures (Figure 23) show the substantial difference between running a rigid tail configuration (CFD only) and a flexible tail (aeroelastic). The first figure shows the buffet simulation using rigid tails at AOA = 34 deg. The next figure shows the same configuration but now for a flexible (aeroelastic) tail. The tail motion causes the vortex breakdown to move upstream, which significantly changes the aerodynamic loading of the tails.

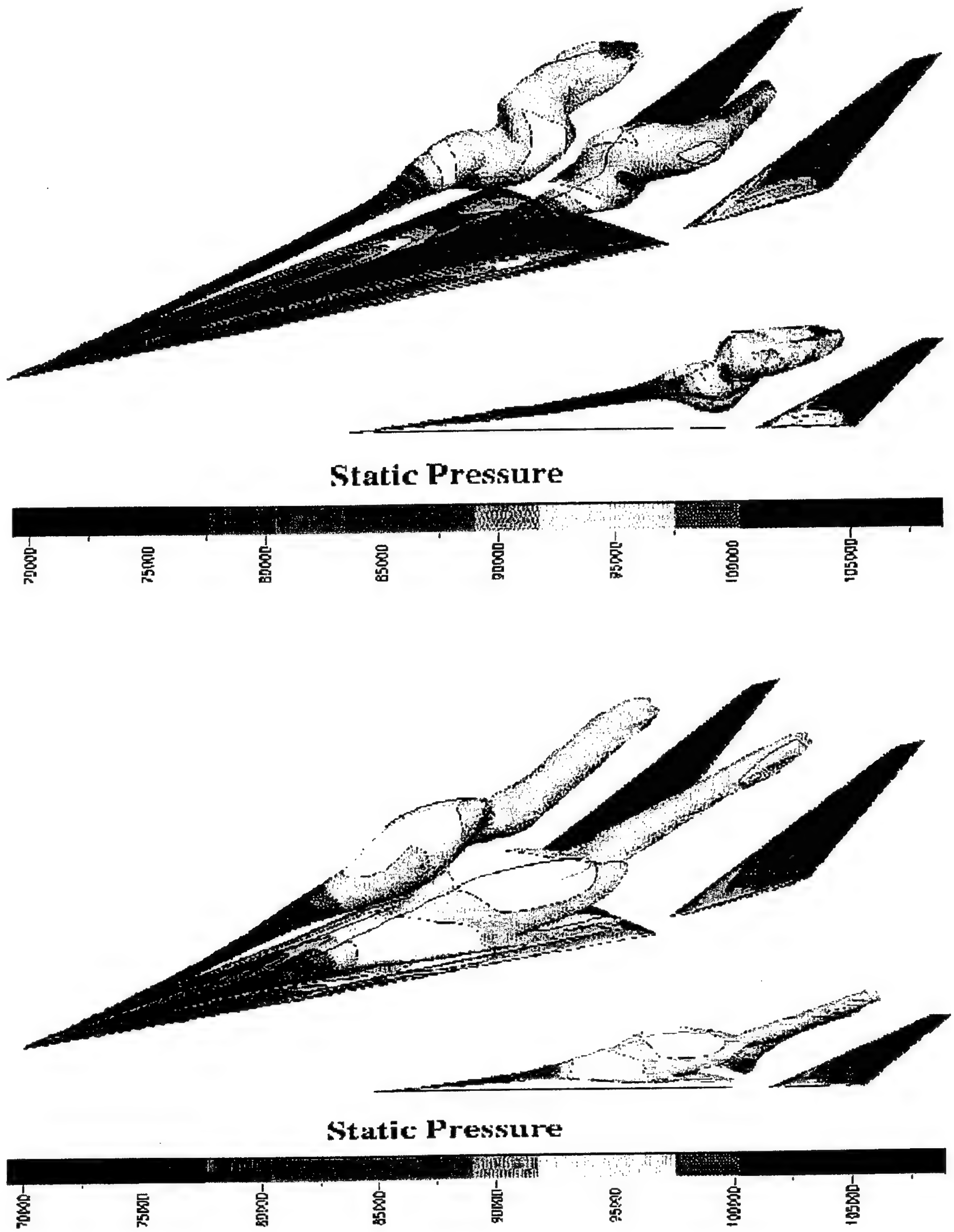


Figure 23. Iso-Valve Surfaces of Total Pressure

Rigid (top) vs. Aeroelastic Simulation (bottom)

Figure 24 shows side-view snapshots of the total pressure iso-surfaces over the configuration model at different angles of attack. The figure shows that at AOA=26 deg. the leading-edge vortices break down behind the twin tail. However, as the angle of attack increases, the vortex-breakdown flow moves upstream ahead of the twin tail. This is the reason behind the increase of the buffet responses at angles of attack higher than 25, as observed by various experimental investigations. The figure also shows increase in the size of the breakdown bubble with the increase of the angle of attack.

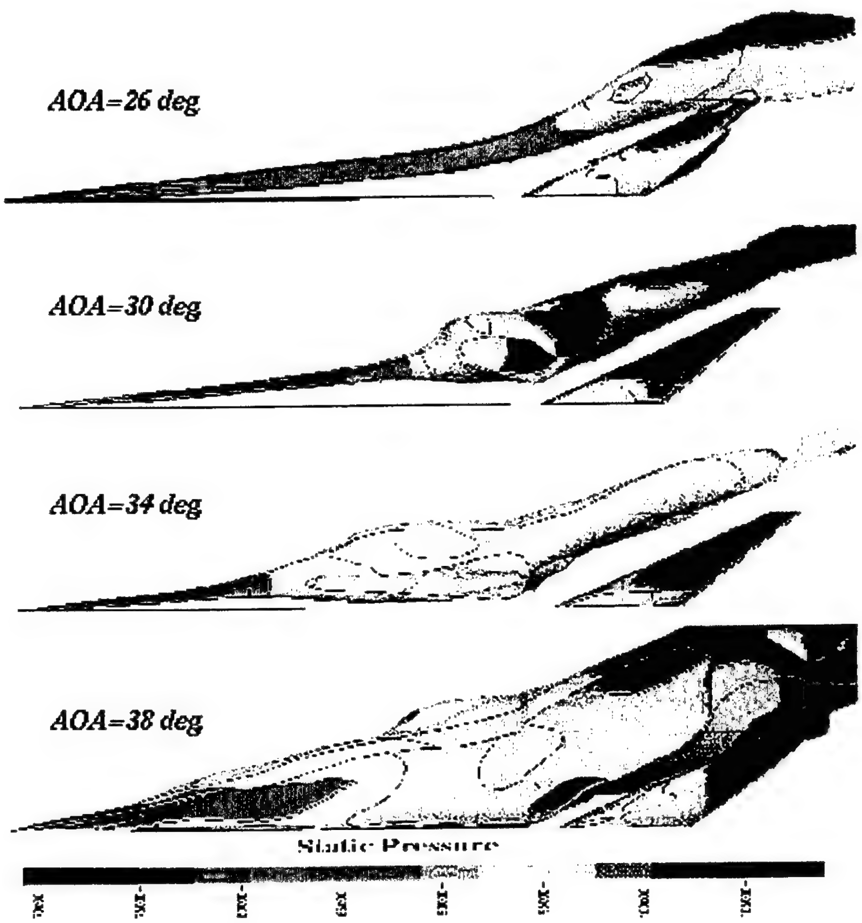


Figure 24. Total Pressure Iso-Surfaces as a Function of Angle of Attack

Aeroelastic Results

Figure 25 shows the histories of the bending and torsion deflections of the left (bottom in figure) and right (top in figure) tail tips at different angles of attack. The figure shows that increasing the angle of attack has led to an increase in both the bending and torsion deflections. The frequency of the torsion deflections is more than twice the frequency of the bending deflections, in agreement with the experimental observations. The figure also shows a slight phase lag in the bending deflections with the increase of the angle of attack. The right and left tails have the same level of deflection. However, they are moving to the outboard direction in an asymmetric manner due to the irregular vibrations of the left and right tails.

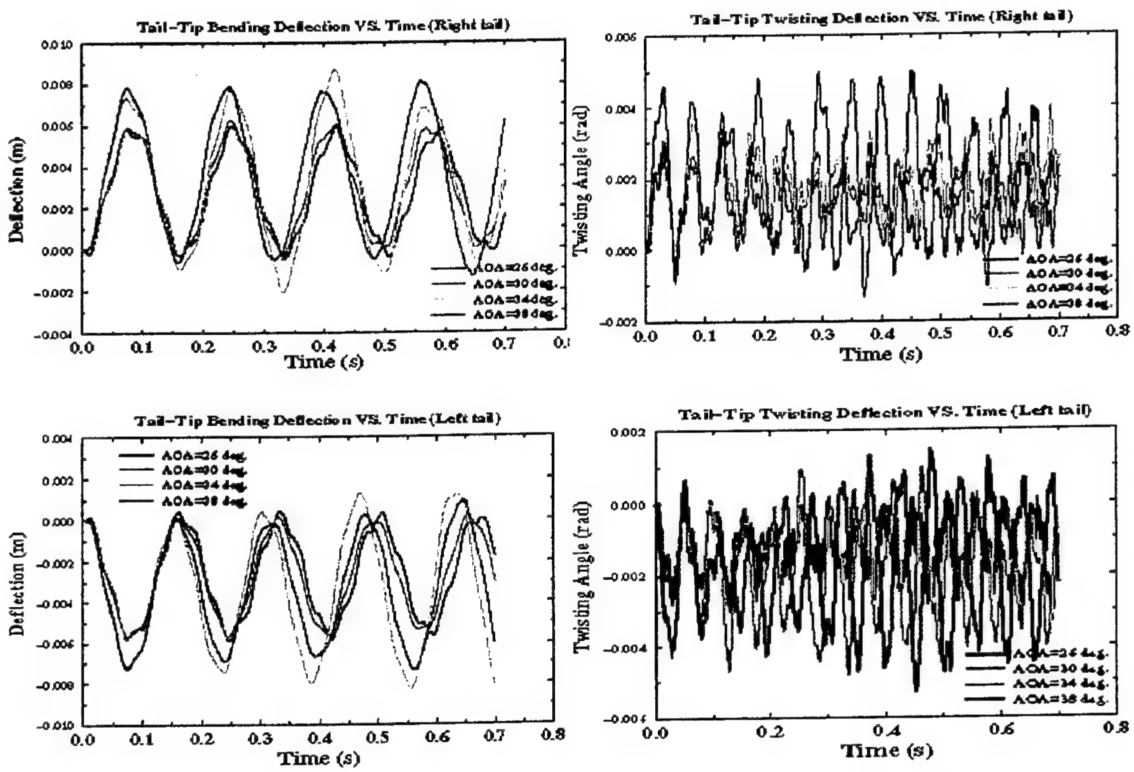


Figure 25. Tail-tip Bending and Twisting Deflection for Left and Right Tail

Figure 26 shows the history of the root bending moment of the right tail at different angles of attack. Positive moments correspond to an outward force on the tail. At AOA=26 deg., there is no apparent variation in the root bending moment. This is because of the absence of vortex breakdown flow in front of the twin tail, as shown in the previous figure. As the angle of attack increases, the root bending moment increases due to the upstream motion of the vortex breakdown flow which causes the unsteady dynamic loads on the tails.

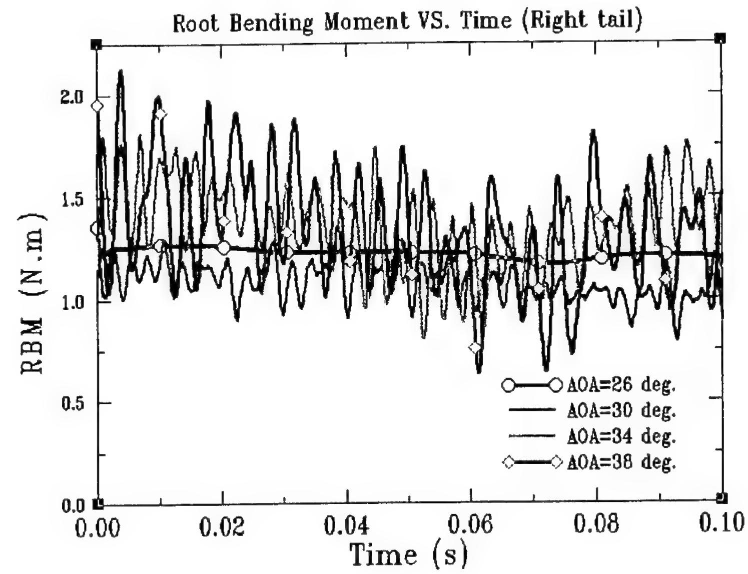


Figure 26. Root Bending Moment for Various Angles of Attack

Validation

Figure 27 shows the mean and RMS values of the right-tail root bending moment coefficients as a function of the angle of attack. The experimental data of Washburn et al.[4] are also shown in the figure. The computed results agree well with the experimental data. At an angle of attack higher than 25, the RMS of the root bending moment increases with the increase of the angle of attack due to the upstream motion of the vortex

breakdown flow in front of the twin tails. Positive moments correspond to an outward force on the tails. The outward bending of the tails is due to the suction pressure caused by the primary vortex which passes outboard of the tails. The effective angle of attack induced by the outward spanwise flow from the primary vortex on the tail also contributes to the outward bending moment.

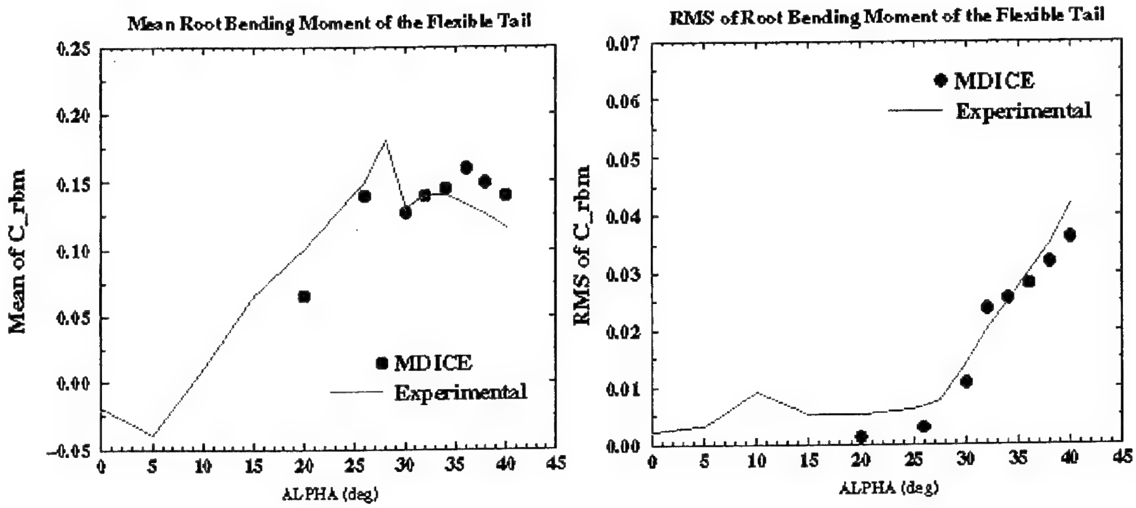


Figure 27. Mean and RMS Valves of the Right Tail Root Bending Coefficients as a Function of the Angle of Attack

Figure 28 shows the distribution of the lift coefficient against the angle of attack. The experimental data of Washburn is also shown in the figure. A maximum of 15% error in the lift coefficient is observed at high angles of attack. This error in the lift coefficient may be attributed to the laminar flow assumption which is not a good approximation at high angles of attack at which massive separation occurs over the wing body. The angle of attack at which a reduction in lift coefficient is first observed is almost at 30.0 degrees. In the experimental data of Washburn, this angle was in the range of 28.5-30.5 degrees.

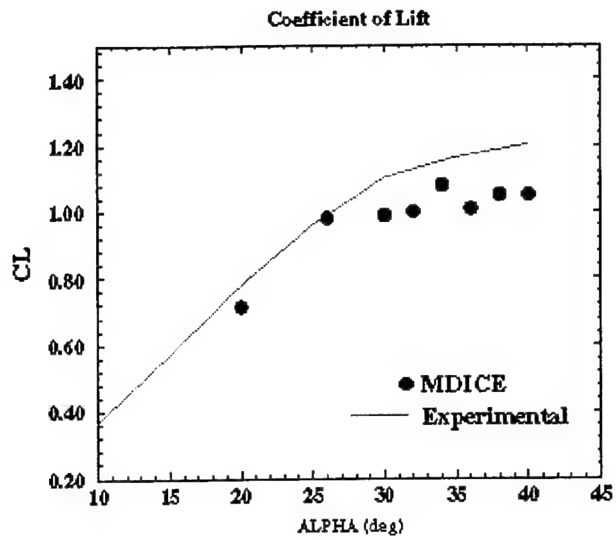


Figure 28. Lift Coefficient as a Function of Angle of Attack

Figure 29 shows the RMS buffet pressure and RMS surface pressures at five transducers on the inner and outer surfaces of the right tail. The buffet pressure is defined as the instantaneous differential pressure across the tail surface, and it is normalized by the free-stream dynamic pressure. The buffet pressures show a sharp increase after 26° angle of attack. The sharp increase in the buffet pressures corresponded with the vortex breakdown position crossing the trailing edge. The RMS buffet pressures were a strong function of transducer location and locations 4 and 5 yielded the greatest levels. In the experimental data of Washburn, location 4 yielded the greatest level. The surface pressure fluctuations were sensitive to the tail side and transducer location. Generally, the inner surface RMS pressure levels were larger than those of the outer surface, in agreement with the experimental observations of Washburn et al.[4]. The lowest RMS pressure levels were observed at transducer location 2. This location corresponds to the transducer furthest from the tail leading edge. In the experimental data of Washburn, locations 2 and 5 show the lowest levels.

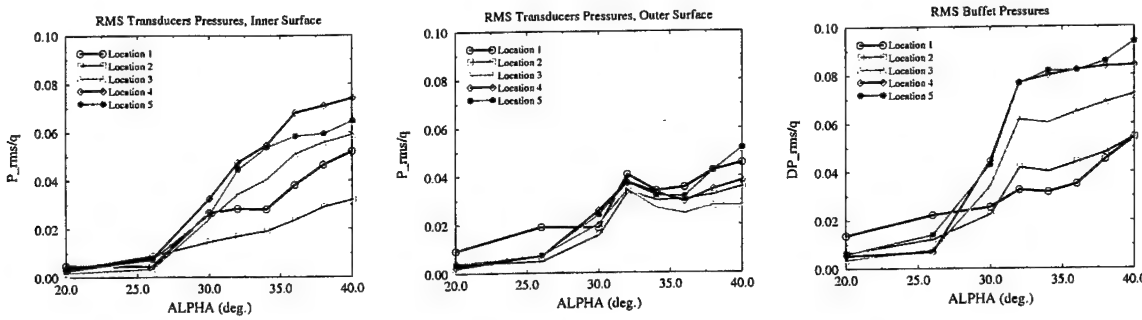


Figure 29. RMS Buffet and Surface Pressures at Five Transducer Locations on the Right Tail

Figure 30 shows the buffet excitation spectra of the tail-tip transducer (50% chord and 90% span) for several angles of attack versus the non-dimensional frequency. The non-dimensional frequency is defined as (n = f b / U), where f is the frequency [Hz], b is the tail span, and U is the uniform velocity. The buffet excitation parameter, is defined as the contribution to power spectrum of in a frequency band and is the best in describing the buffet excitation level. The figure shows that at angles of attack of 30 and higher the buffet excitation parameter increased sharply due to the upstream movement of the vortex breakdown ahead of the twin-tail. Generally, there were two distinct frequency peaks in the frequency band. These peaks represent coherent fluctuations in the flow at those frequencies.

Figure 31 shows the variation of the first two dominant non-dimensional frequencies of the tail-tip transducer versus the angle of attack. The experimental data of Washburn et al.[4] is also shown in the figure. The results are in very good agreement with the experimental data. The frequency peaks shift to a lower frequency as the angle of attack increases. The first two frequency peaks are moderately close to each other, which indicate that the pressure field contains energy over a narrow frequency band. This is in

agreement with the observations of Washburn[4] and Martin and Thompson[5]. The general reduction in frequency of the vortex breakdown induced pressure field, as angle of attack increases, has also been observed on the F/A-18 and on different delta wings. The F/A-18 tail pressure excitation spectra, however, generally exhibit only one peak.

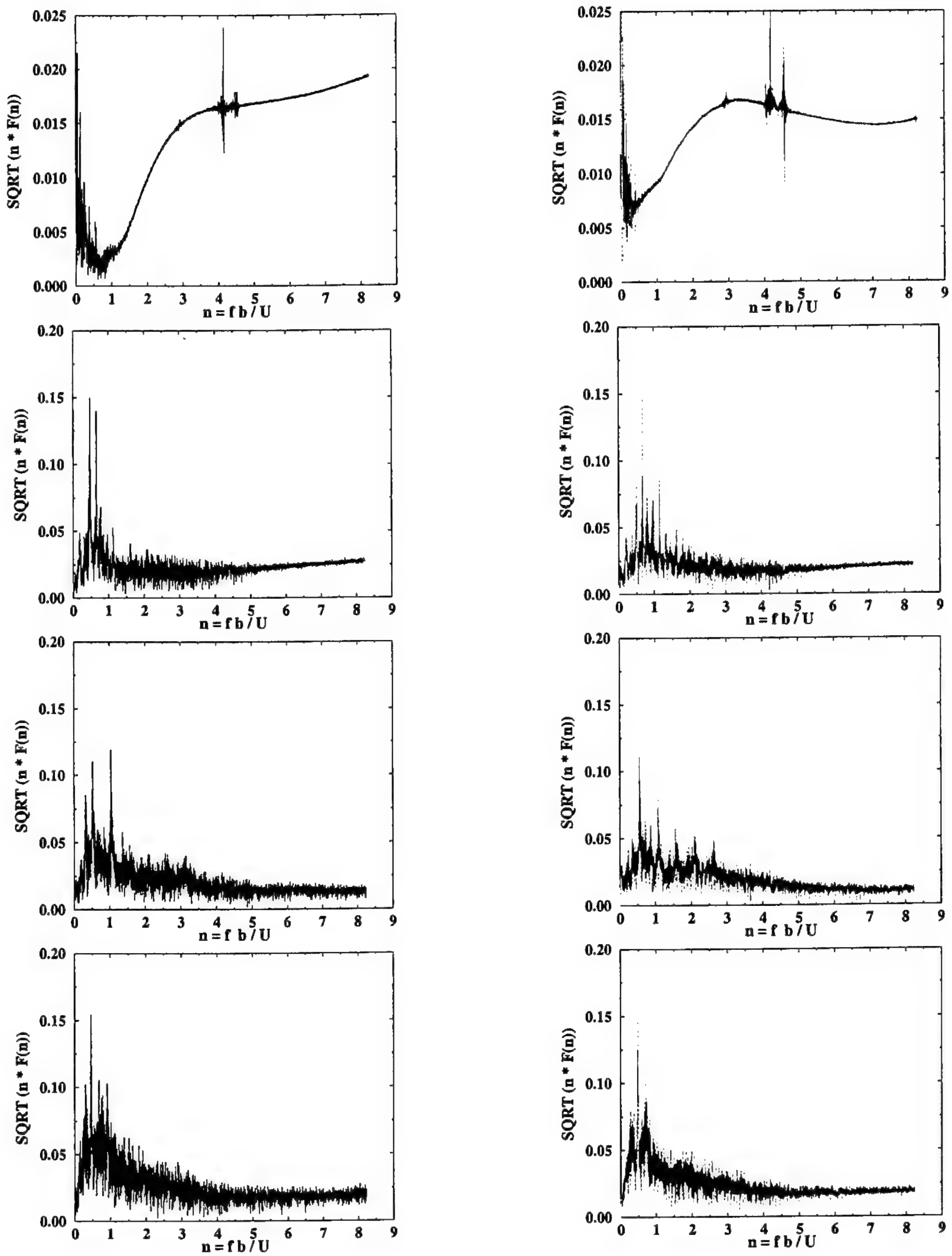


Figure 30. Buffet Excitation Spectra

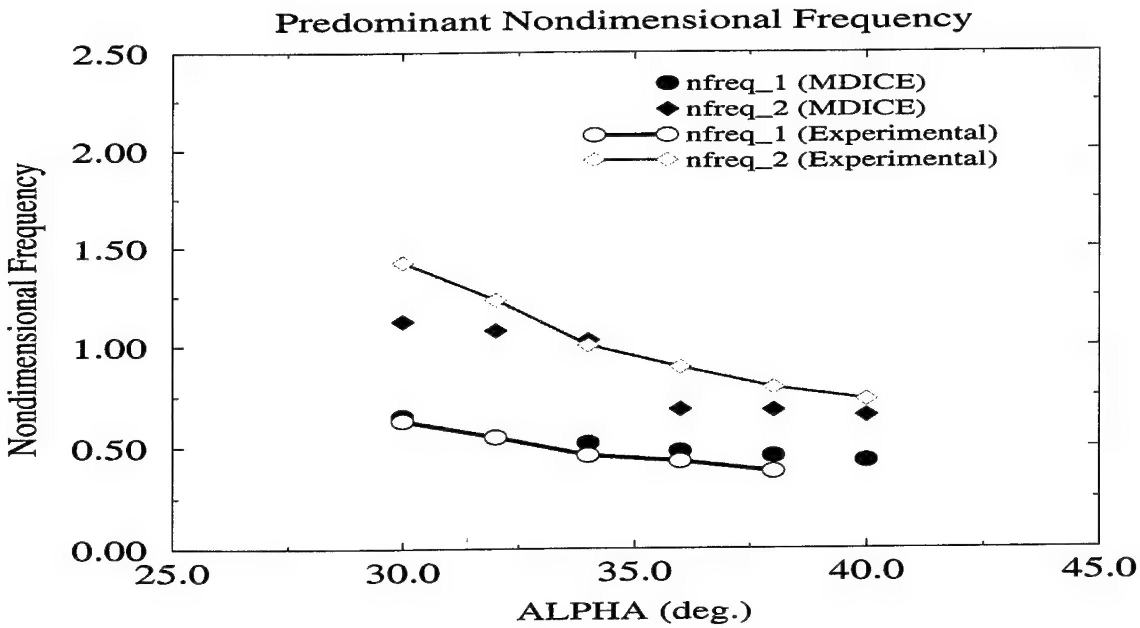


Figure 31. First Two Dominant Non-Dimensional Frequencies as a Function of Angle of Attack

Conclusions

In summary, the following conclusions can be made from this study.

- A very good agreement with the experimental data was achieved.
- A sharp increase in the buffet pressures was observed as the vortex breakdown crossed the wing trailing edge.
- The frequency of the torsion deflections was almost twice that of the bending deflections.
- There is a slight phase lag in the bending deflections with the increase of the angle of attack.

5. COLLABORATION WITH AEROSPACE COMPANIES AND UNIVERSITIES

During the course of this project we have been collaborating with several organizations. After the start of this contract, CFD Research awarded a subcontract to Northrop Grumman, Military Advanced Systems Division (MASD), in Pico Rivera, CA to participate in this contract. In the third year of the contract, Lockheed Martin Tactical Aircraft Systems, Forth Worth, TX, started to collaborate on this effort with their own funding. In this chapter a brief overview of both collaborations is given.

5.1 Northrop Grumman

Northrop Grumman was selected as a partner because they had a similar proposal and vision of what needed to be done in this contract. CFDRC and Northrop Grumman have mutually agreed upon the following work plan.

5.1.1 Work Plan

1. Install and Evaluate MDICE (first beta release) and various CFDRC modules
2. Integrate Northrop's flow solver GCNSfv into MDICE
3. Integrate Northrop's Linear Structural Solver into MDICE
4. Modify grid solver module and incorporate it into MDICE
5. Set up CFD and FEM models for AGARD 445.6 wing
6. Establish interface between models, and run static aeroelastic case.

7. Perform NASTRAN linear aeroelastic analysis for comparison.
8. Set up CFD and FEM models for B2 configuration
9. Run static B2 configuration, including inlets but with no control surface deflection.
10. Install and evaluate second beta release of MDICE and repeat steps 2-9.

During the third year of the program, we have added one more task to this program. This task was the creation of a report entitled "Generic Flight Control System for MDICE: Desired Capabilities & Survey of Candidates for Implementation".

5.1.2 Results

Northrop Grumman has integrated their own flow solver GCNSfv with MDICE, as well as their own structural solver, fluid-structure interface module (based on Brown [2]), and their own grid movement module. The flow solver was parallelized. Initially there was no explicit support for parallel modules in MDICE, and the programmer had to do a lot of extra work to obtain a parallel capability. During the second year of the project, explicit support for parallel modules was added to MDICE.

They have set up and performed steady state aeroelastic simulations on the AGARD 445.6 wing and the B2 bomber with good validation results. The actual results on the B2 are proprietary to Northrop Grumman. The work was presented at the second MAPINT 98 / MDICE Workshop, August 25-27, 1998 [3]. From this presentation, the last two summary slides are given verbatim.

Feedback on Using MDICE

- Example codes/problems are helpful for development
- MDICE API is very good (minimal and complete for a broad range of CFD/structural methodologies)
- MDICEing a code requires significant understanding of how MDICE functions
- MDICE itself seems very reliable

Feedback on MDICE Project

- Impressed by collection of technologies (object libraries, parsed command language, distributed computing models, useful GUI)
- Practical concepts, reliable implementation – good prospects for industry use
- Would like to see continued high level object library development
- Would like to see full consideration of distributed computing issues (e.g. scalability, portability)

Furthermore it was reported that the total level of effort at Northrop Grumman was 18 weeks and the steep part of the learning curve was about 2 weeks.

5.2 Lockheed Martin

During the third year of the contract, Lockheed Martin Tactical Aircraft Systems has collaborated with CFDRC on this project. They have integrated their own Cartesian grid flow solver (Splitflow) and their own influence coefficient solver module with MDICE.

The nature of the work with Lockheed Martin is proprietary. In summary, they have been working on wing-body configurations with multiple fluid-structure interfaces present (i.e. utilize MDICE's multi-patching interfaces). Without going into the details, their feedback has made a significant contribution to the MDICE project and we would like to thank them for their testing and validation efforts.

6. RECOMMENDATIONS FOR FUTURE WORK

MDICE has set a standard for creating an operating system for engineering analysis. Once applications are MDICE compliant, those applications can exchange data and interoperate with each other. Under this contract, the MDICE framework has been focused on aerodynamic and structural interaction. However, with some extensions and modifications, many more applications are possible. In this section the recommendations for future work are summarized.

- More fluid-structure interface algorithms could be integrated with MDICE. In particular, CFDRC plans to incorporate another fluid-structure interfacing method based on the Boundary Element Method (BEM). This method has been developed by Zona Technologies under a Phase II contract with NASA Langley.
- MDICE facilitates several methods for fluid structure interfacing. It is unclear what the relative merits are for each interfacing method. Therefore a study needs to be conducted which assesses the performance of each method for a variety of configurations and conditions (both steady state and transient simulations). Such a study could facilitate the definition of clear guidelines on what method to select for a certain application.
- Add generic flight controller technology to MDICE. This would result in an aero-servo-elastic simulation capability. This could include specialized modules for trim and 6DOF analysis. In combination with the Chimera module, the 6DOF module could be used for store separation simulations. A generalized capability such as

available in the MATLAB program could be provided as well by means of a direct interface to MATLAB. As an extra benefit, the MATLAB-provided neural net based controllers could then be used in combination with MDICE simulations.

- MDICE has sufficient infrastructure for performing multi-disciplinary optimization problems. Integrating an optimization module with MDICE would provide such a capability. An example of this application is the Boeing MDOPT (Multi-Disciplinary OPTimization) project.
- The integration of a non-linear structural analysis module (e.g. CFDRC's FEMSTRESS) is important for a variety of applications, e.g. modeling piezoelectric controlled structures.
- Provide an aero-thermal or aero-thermal-elastic coupling in MDICE. Thermoelasticity capabilities are required in the MDICE environment to study aeroelastic problems at hypersonic speeds and in internal combustion engines and fluid-structure interaction in propulsion systems (automotive application, turbomachinery, turbo jet engines, ...)
- High Order Accurate CFD: Some of the more sophisticated aeroelastic problems require very high accurate CFD techniques. Fifth order and seventh order accurate CFD modules should be used in problems like Blade Vortex Interaction (BVI) in rotorcraft applications. Noise prediction and noise induced vibration are other examples. The lack of integration of high order accurate CFD modules and structural analysis modules is the reason that research in these areas are not yet matured enough. Thus, the addition of high order accurate CFD module in MDICE environment is highly required for the prediction and control of these very important

phenomena. The high order accurate CFD module could be the CFDRC developed 7^{th} order accurate module or the recently developed 5^{th} order accurate of OVERFLOW (NASA ARC).

- Provide a better MSC/NASTRAN interface. The current interface is file-based and is suitable for steady state simulations. MSC/NASTRAN has been used in combination with MDICE for modal and FEM analysis (steady state aeroelastic). Currently, for transient simulations, MacNeal Schwendler recommends to use a separate solver which needs to be compatible with the NASTRAN file format. The MDICE structural solver can be used for this purpose.

Throughout this report the various planned additions to MDICE have been indicated. It is envisioned that there will be a continued need for more engineering code integration efforts and further improvements to MDICE. This means that the software will evolve over time. Furthermore, issues such as software maintenance, technical user support, and training are of outmost importance for effective use of MDICE by a variety of organizations and companies. CFDRC is fully committed to provide all necessary MDICE support to those organizations and companies.

7. REFERENCES

[1] Smith, M.J., Hodges, D.H., Cesnik, C.E.S., "An Evaluation of Computational Algorithms to Interface Between CFD and CSM Methods," WL_TR-96-3055, November 1995.

[2] Brown, S.A., "Displacement Extrapolation for CFD and CSM Aeroelastic Analysis," AIAA Paper 97-1090, 1997.

[3] Blosch, E.L., "Progress at NGC Using CFDRC's MDICE for Aeroelastic Analysis," presentation material at MAPINT 98/MDICE Workshop, August 25-27, 1998 at WPAFB.

[4] Washburn, A. E., Jenkins, L. N. and Ferman, M. A., "Experimental Investigation of Vortex-Fin Interaction," AIAA 93-0050, AIAA 31st ASM, Reno, NV, January 1993.

[5] Martin, C. A. and Thompson, D. H., "Scale Model Measurements of Fin Buffet due to Vortex Bursting on F/A-18," AGARD-CP-497, May 1991.

8. MDICE BIBLIOGRAPHY

1. Sheta, E. F., Harrand, V. J., Thompson, D. E., and Strganac, T. W., "Computational and Experimental Investigation of Limit Cycle Oscillations in Nonlinear Aeroelastic Systems" to be published in the *AIAA Journal of Aircraft*, Log No. C-9681, 2001.

2. Sheta, E. F., Harrand, V. J., and Huttsell, L. J., "Active Vortical Flow Control for Alleviation of Twin-Tail Buffet of Generic Fighter Aircraft," to be published in the *Journal of Fluids and Structures*, Log No. JPG/00/02, 2001.

3. Sheta, E. F. and Huttsell, L. J., "Characteristics of Vertical Tail Buffet of F/A-18 Aircraft," to be published in the 39th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, January 8-11, 2001.

4. Sheta, E. F., Rock, S. G. and Huttsell, L. J., "Vortical Flow Structure of F/A-18C Aircraft at High Angles of Attack," to be published in the 39th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, January 8-11, 2001.

5. Sheta, E. F., "Control of Twin-Tail Buffet Responses using Normal Leading-Edge Blowing," AIAA 2000-4227, 18th AIAA Applied Aerodynamic Conference, Denver, CO, August 14-17, 2000.

6. Sheta, E. F. and Strganac, T. W., "A Computational and Experimental Investigation of Transient Nonlinear Viscous Aeroelastic Effects in Fixed Wing Vehicles," Final Report, STTR Phase I, Contract No. F49620-99-C-0045, Air Force Office of Scientific Research, Arlington, VA, April 2000.

7. Sheta, E. F., "Prediction and Control of Twin-Tail Buffet of Fighter Aircraft," Final Report, SBIR Phase I, Contract No. F33615-99-C-3213, Air Force Research Laboratory, Dayton, OH, February 2000.

8. Sheta, E. F., Harrand, V. J., Thompson, D. E., and Strganac, T. W., "Computational and Experimental Investigation of Limit-Cycle Oscillations in Nonlinear Aeroelastic Systems," AIAA 2000-1399, 41st AIAA/ ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Atlanta, GA, April 3-6, 2000.

9. Sheta, E. F., Harrand, V. J., and Huttsell, L. J., "Active Vortical Flow Control for Alleviation of Twin-Tail Buffet of Generic Fighter Aircraft," AIAA 2000-0906, 38th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, January 10-13, 2000.

10. Sheta, E.F., Harrand, V.J., Strganac, T., "Computational and Experimental Investigation of Limit Cycle Oscillations in Non-linear Aeroelastic Systems," to be published in: Structures, Structural Dynamics, and Materials Conference, April 2000.

11. Sheta, E.F., Harrand, V.J., Huttsell, L.J., "Active Vortical Flow Control for Alleviation of Twin-Tail Buffet of Generic Fighter Aircraft," AIAA 2000-0906, Jan 2000, Reno, NV.

12. Sheta, E. F., Siegel, J. M., Golos, F. N., and Harrand, V. J., "Twin-Tail Buffet Simulation Using a Multi- Disciplinary Computing Environment, '' NASA/CP-1999-209136, CEAS/AIAA/ICASE/NASA Langley International Forum in Aeroelasticity and Structural Dynamics, Williamsburg, Virginia, June 22- 25, 1999.

13. Siegel, J. M., Jr., Parthasarathy, V., Kingsley, G. M., Dionne, P. J., Harrand, V. J. and Luker, J. J., "Application of A Multi- Disciplinary Computing Environment

(MDICE) for Loosely Coupled Fluid-Structural Analysis," AIAA-98-4866, 7th AIAA/USAF/NASA/ISSMO Symposium of Multidisciplinary Analysis and Optimization, September 1998.

14. Kingsley, G. M., Siegel, J. M., Harrand, V. J., Lawrence, C., and Luker, J., "Development of the Multi- Disciplinary Computing Environment (MDICE)," AIAA-98-3857, 7th AIAA/USAF/NASA/ISSMO Symposium of MDO, September 1998.

15. Harrand, V. J., Siegel, J. M., Singhal, A. K., and Whitmire, J. B., "Key Components and Technologies for the NCC Computing Framework," 34th AIAA/ASME/SAE/ASEE JPC Conference, AIAA Paper 98-3857, July 1998.

16. Kingsley, G.M., Harrand, V.J., and Lawrence, C., "A Visual Computing Environment for Computational Aerosciences," Pro-ceedings of the 1996 Computational Aerosciences Workshop, NASA Ames Research Center, pp. 313-318, August 1996.

17. Coirier, W.J., Golos, F.N., Harrand, V.J., Przekwas, A.J., "CFD-DTF: A Data Transfer Facility for CFD and Multi-Disciplinary Analysis," AIAA Paper 98-0125, 36th Aerospace Sciences Meeting and Exhibit, 1998.

18. Cole and Gary, "VCE Application at NASA Lewis: Inlet-Engine Simulation," Proceedings of the First VCE Workshop, NASA Research Center, November 1997.

19. Liu, Nan-Suey, "VCE Application to National Combustion Code," Proceedings of the First VCE Workshop, NASA Lewis Research Center, November 1997.

20. Edwards, David, "VCE Application at Pratt & Whitney," Proceedings of the First VCE Workshop, NASA Lewis Research Center, November 1997.

21. Sandel, Wolfgang, "VCE Applications at AlliedSignal Engines," Proceedings of the First VCE Workshop, NASA Lewis Research Center, November 1996.